LABASSIGNMENT-19.2

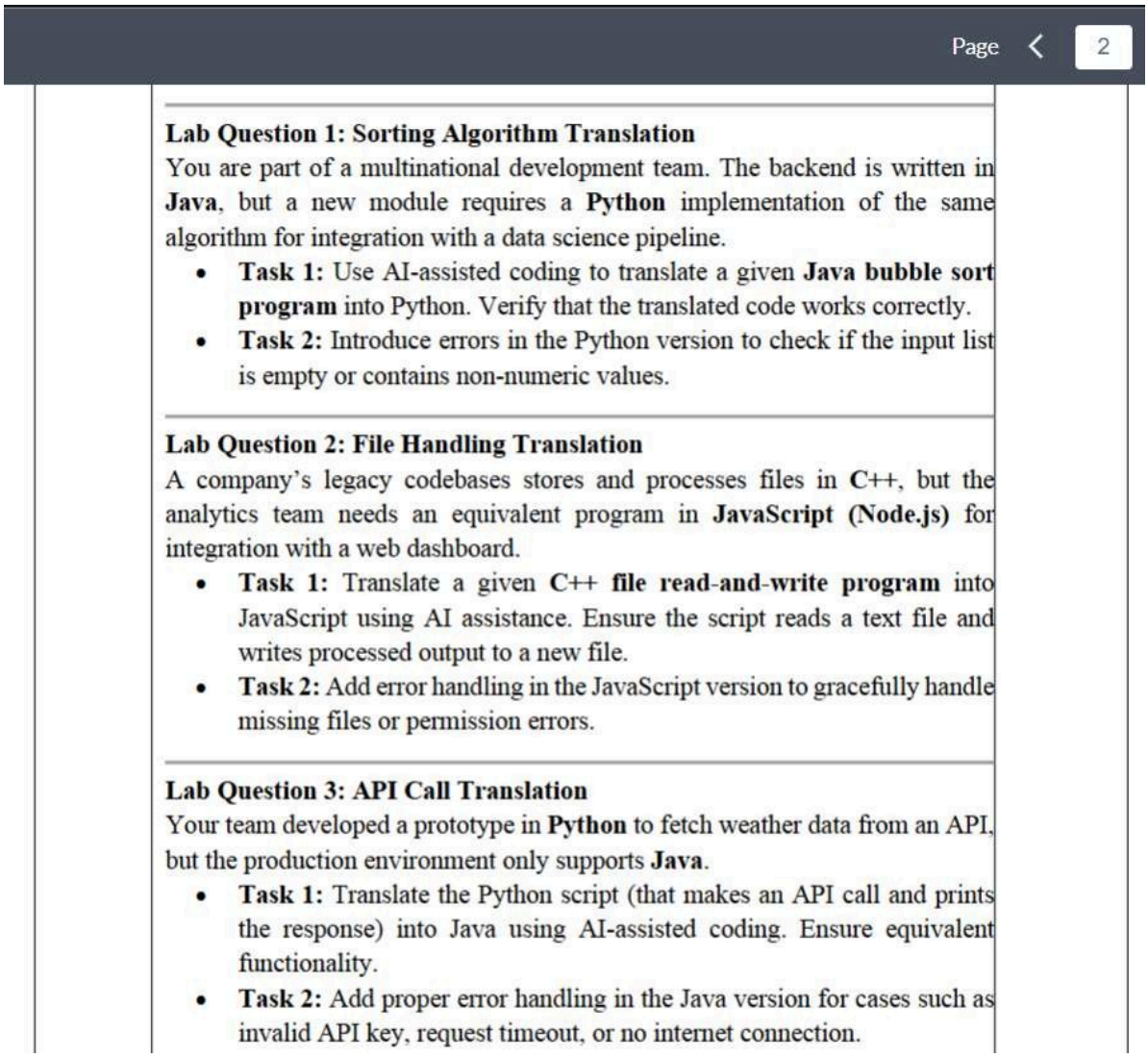
NAME:B.SRINITHI

RoLLNO:2403A51413

COURSE:AIASSISTEDCODING

BATCH:01(AIML)

QUESTIONS:



Lab Question 1: Sorting Algorithm Translation

You are part of a multinational development team. The backend is written in **Java**, but a new module requires a **Python** implementation of the same algorithm for integration with a data science pipeline.

- **Task 1:** Use AI-assisted coding to translate a given **Java bubble sort program** into Python. Verify that the translated code works correctly.
- **Task 2:** Introduce errors in the Python version to check if the input list is empty or contains non-numeric values.

Lab Question 2: File Handling Translation

A company's legacy codebases stores and processes files in **C++**, but the analytics team needs an equivalent program in **JavaScript (Node.js)** for integration with a web dashboard.

- **Task 1:** Translate a given **C++ file read-and-write program** into JavaScript using AI assistance. Ensure the script reads a text file and writes processed output to a new file.
- **Task 2:** Add error handling in the JavaScript version to gracefully handle missing files or permission errors.

Lab Question 3: API Call Translation

Your team developed a prototype in **Python** to fetch weather data from an API, but the production environment only supports **Java**.

- **Task 1:** Translate the Python script (that makes an API call and prints the response) into Java using AI-assisted coding. Ensure equivalent functionality.
- **Task 2:** Add proper error handling in the Java version for cases such as invalid API key, request timeout, or no internet connection.

LABQUESTION1:

PROMPT:

Youarepartofamultinationaldevelopmentteam. ThebackendsystemusesJava,butanewmodulerequiresa Python implementation for integration with a data science pipeline.

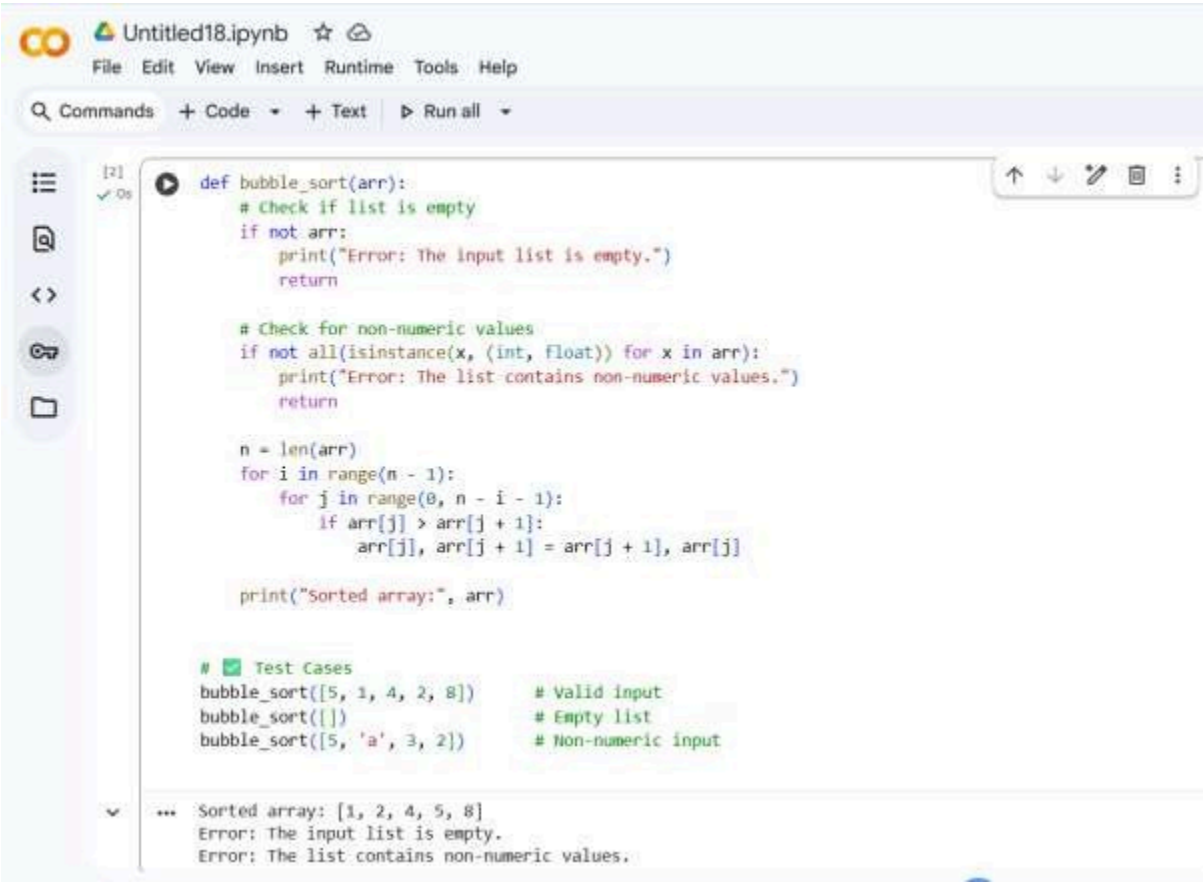
Task1: UseAI-assistedcodingtotranslatethefollowingJava Bubble Sort program into Python and verify that it works correctly.

Task 2: Modify the Python version to include error handling thatchecksiftheinputlistisemptyorcontainsnon-numeric values. If so, print an appropriate error message instead of sorting.

CODE:

```
publicclassBubbleSort{
  publicstaticvoidmain(String[]args){ in
    t[] arr = {5, 1, 4, 2, 8};
    bubbleSort(arr);
    System.out.println("Sorted array:");
    for (int num : arr) {
      System.out.print(num+"");
    }
  }

  staticvoidbubbleSort(int[]arr){ i
    nt n = arr.length;
```

```
    for(inti=0;i<n -1;i++){
       for(intj=0;j<n-i-1;j++){       if
          (arr[j] > arr[j + 1]) {
             inttemp = arr[j];
             arr[j]  =arr[j  +1];
             arr[j+1]= temp;
          }
       }
    }
  }
}
```



LABQUESTION2:

PROMPT:

Youarepartofadevelopmentteamthatismodernizinga company's legacy C++ system. Theoldsystemreadsatextfile,processesitscontent,and writes the results to a new file.

Task1: UseAI-assistedcodingtotranslatethegivenC++file read-and-write program into JavaScript (Node.js). EnsurethattheJavaScriptversioncorrectlyreadsdatafrom a textfileandwritesprocessedoutput(forexample,uppercase text) into a new file.

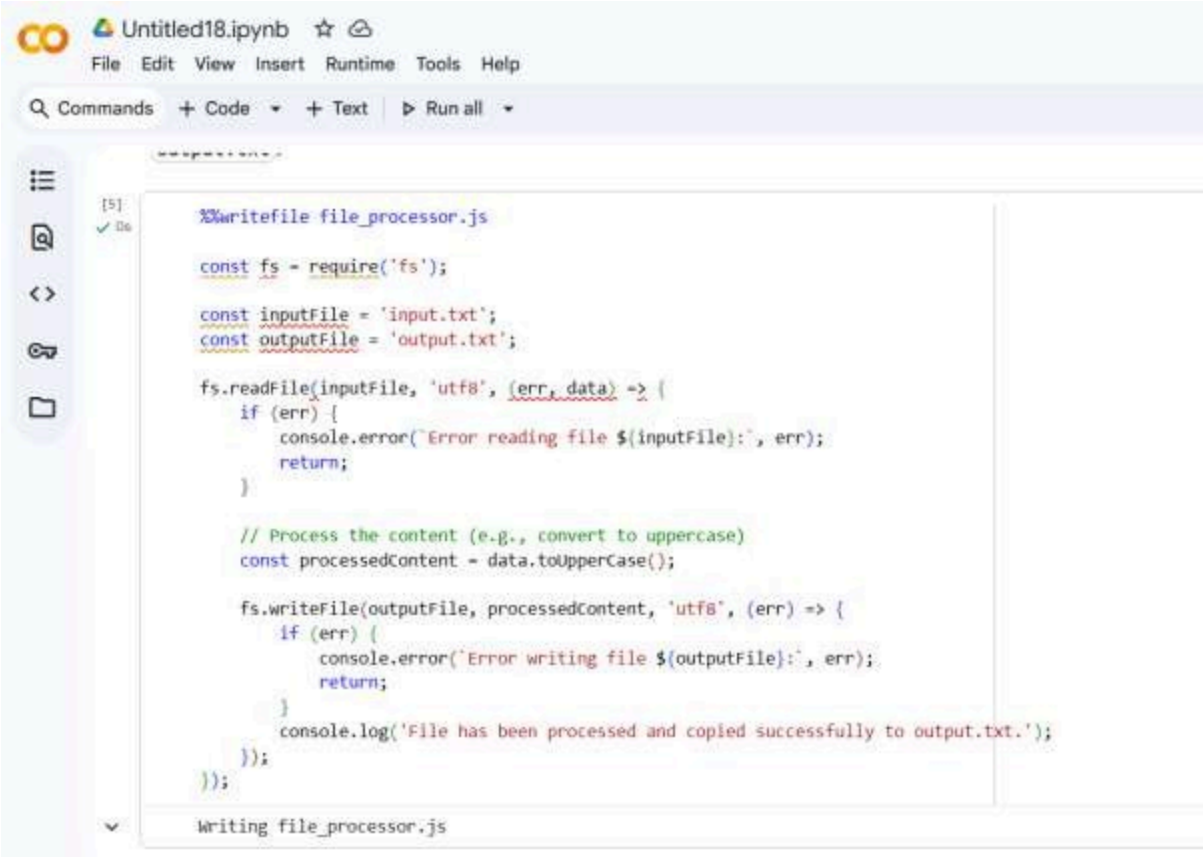
Task2

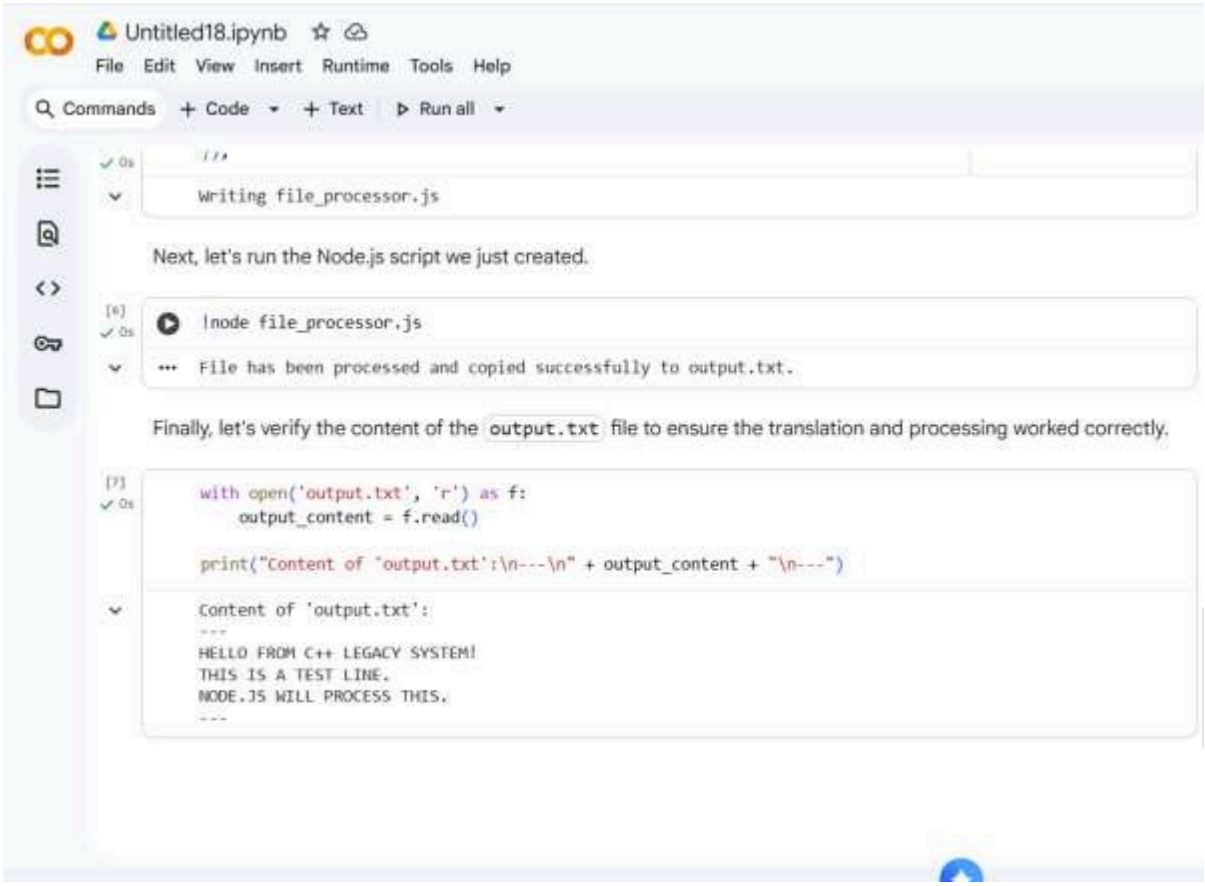
ExtendtheJavaScript(Node.js)file-handlingprogramby adding **error handling**. Theupdatedscriptshouldhandlemissinginputfiles,invalid permissions, or any I/O errors gracefully, showing user-friendlymessageswithoutcrashingtheprogram.

CODE :

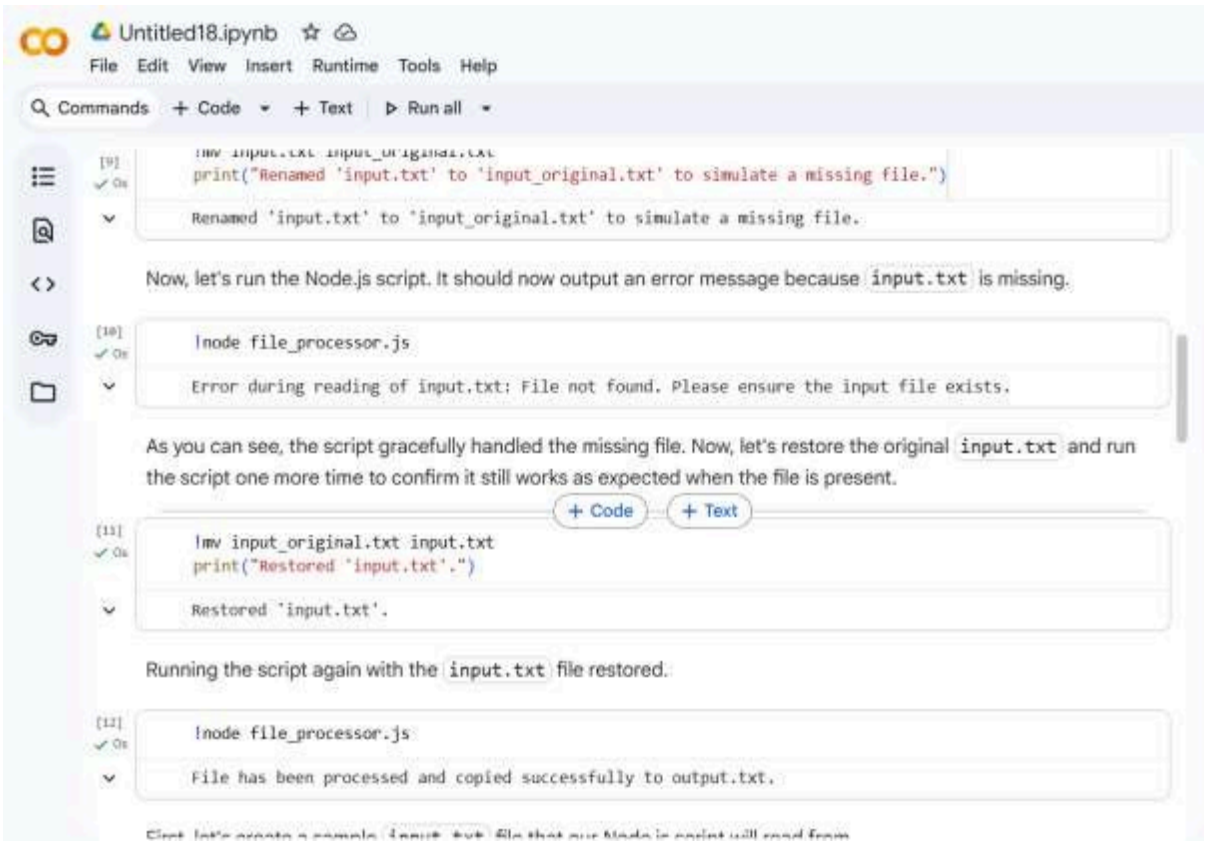
```
#include<iostream
> #include
<fstream>
#include <string>
usingnamespacestd;

intmain(){
   ifstream inputFile("input.txt");
   ofstream outputFile("output.txt");
   string line;

   if(inputFile.is_open()&&outputFile.is_open()){ wh
      ile (getline(inputFile, line)) {
         outputFile<<line<<endl;
      }
      cout<<"Filehasbeencopiedsuccessfully."<<endl;
   }else{
      cout<<"Erroropeningfile."<<endl;
   }

   inputFile.close();
```

```
    outputFile.close();

    return0;
}
```




Untitled18.ipynb
File Edit View Insert Runtime Tools Help
Commands + Code + Text Run all
Writing file_processor.js
Next, let's run the Node.js script we just created.
!node file_processor.js
File has been processed and copied successfully to output.txt.
Finally, let's verify the content of the output.txt file to ensure the translation and processing worked correctly.
with open('output.txt', 'r') as f:
output_content = f.read()
print("Content of 'output.txt':\n---\n" + output_content + "\n---")
Content of 'output.txt':

HELLO FROM C++ LEGACY SYSTEM!
THIS IS A TEST LINE.
NODE.JS WILL PROCESS THIS.


Untitled18.ipynb
File Edit View Insert Runtime Tools Help
Commands + Code + Text Run all
print("Renamed 'input.txt' to 'input_original.txt' to simulate a missing file.")
Renamed 'input.txt' to 'input_original.txt' to simulate a missing file.
Now, let's run the Node.js script. It should now output an error message because input.txt is missing.
!node file_processor.js
Error during reading of input.txt: File not found. Please ensure the input file exists.
As you can see, the script gracefully handled the missing file. Now, let's restore the original input.txt and run the script one more time to confirm it still works as expected when the file is present.
+ Code + Text
!mv input_original.txt input.txt
print("Restored 'input.txt'.")
Restored 'input.txt'.
Running the script again with the input.txt file restored.
!node file_processor.js
File has been processed and copied successfully to output.txt.

Untitled18.ipynb

File Edit View Insert Runtime Tools Help

Commands + Code + Text Run all

First, let's create a sample `input.txt` file that our Node.js script will read from.

[4]
```
with open('input.txt', 'w') as f:
    f.write('Hello from C++ legacy system!\n')
    f.write('This is a test line.\n')
    f.write('Node.js will process this.')

print("Created 'input.txt' successfully.")
```

```
Created 'input.txt' successfully.
```

Now, here is the JavaScript (Node.js) translation of your C++ program. It uses Node.js's built-in `fs` module to handle file operations. I've also added a step to convert the content to uppercase before writing it to `output.txt`.

[3]
```
%%writefile file_processor.js

const fs = require('fs');

const inputFile = 'input.txt';
const outputFile = 'output.txt';

// Function to handle errors gracefully with more specific messages
function handleError(err, operation, filePath) {
    let errorMessage = `Error during ${operation} of ${filePath}: `;
    if (err.code === 'ENOENT') {
        errorMessage += 'File not found. Please ensure the input file exists.';
```

Untitled18.ipynb

File Edit View Insert Runtime Tools Help

Commands + Code + Text Run all

[8]
```
// Function to handle errors gracefully with more specific messages
function handleError(err, operation, filePath) {
    let errorMessage = `Error during ${operation} of ${filePath}: `;
    if (err.code === 'ENOENT') {
        errorMessage += 'File not found. Please ensure the input file exists.';
    } else if (err.code === 'EACCES') {
        errorMessage += 'Permission denied. Please check file permissions.';
    } else {
        errorMessage += `An unexpected I/O error occurred: ${err.message}`;
    }
    console.error(errorMessage);
}

fs.readFile(inputFile, 'utf8', (err, data) => {
    if (err) {
        handleError(err, 'reading', inputFile);
        return;
    }

    // Process the content (e.g., convert to uppercase)
    const processedContent = data.toUpperCase();

    fs.writeFile(outputFile, processedContent, 'utf8', (err) => {
        if (err) {
            handleError(err, 'writing', outputFile);
            return;
        }
        console.log('File has been processed and copied successfully to output.txt.');
    });
});
```

LABQUESTION3:

PROMPT:

API CallTranslation

Youarepartofasoftwaredevelopmentteammigratinga data pipeline from Python to Java. ThePythonprototypemakesanAPIcalltofetchweather data and prints the response.

Task1: UseAI-assistedcodingtotranslatethefollowing PythonAPIcallprogramintoJavawhileensuringequivalent functionality (fetch data and print the JSON response).

Task2

ExtendthetranslatedJavaAPIprogramtoinclude**robust error handling**. TheJavaversionshouldgracefullyhandlethefollowing scenarios:

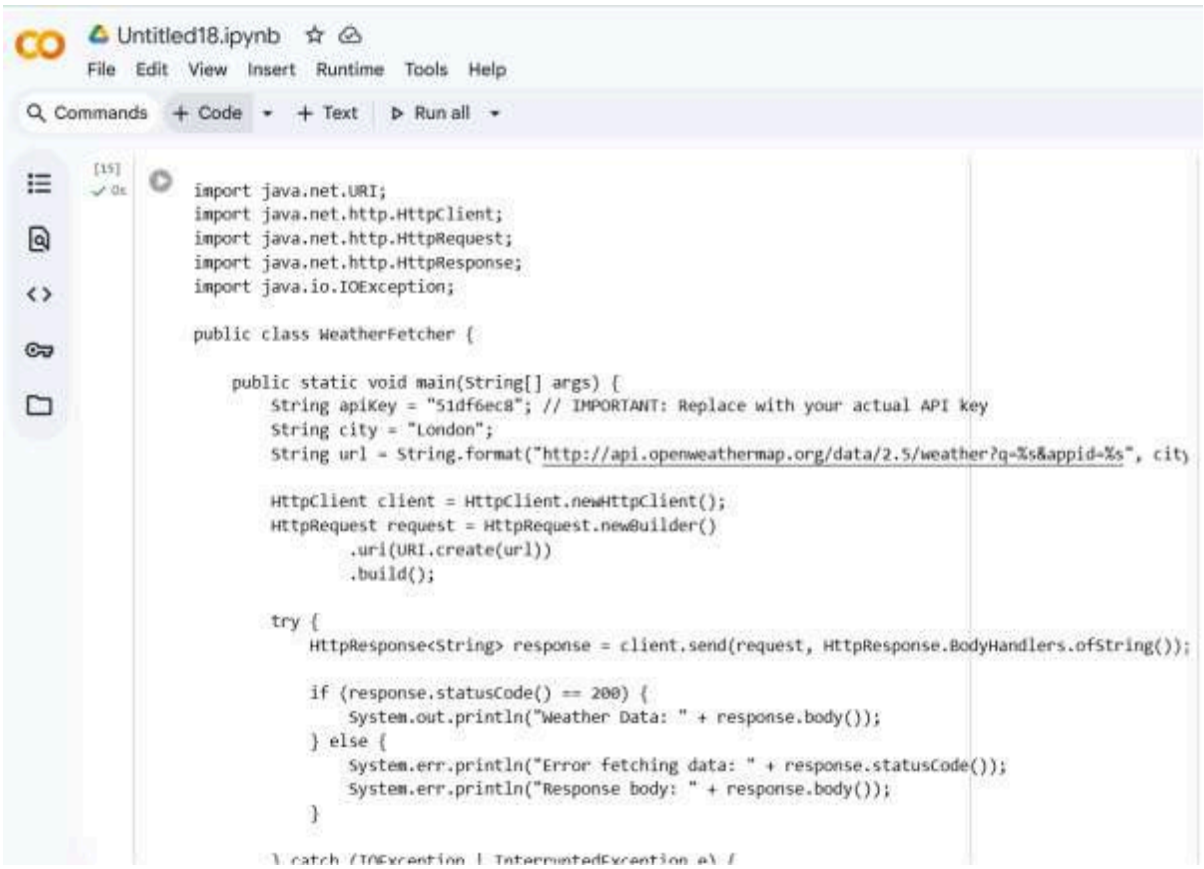
- InvalidAPIkey(HTTP401)
- Requesttimeout
- Nointernetconnectionornetworkissues

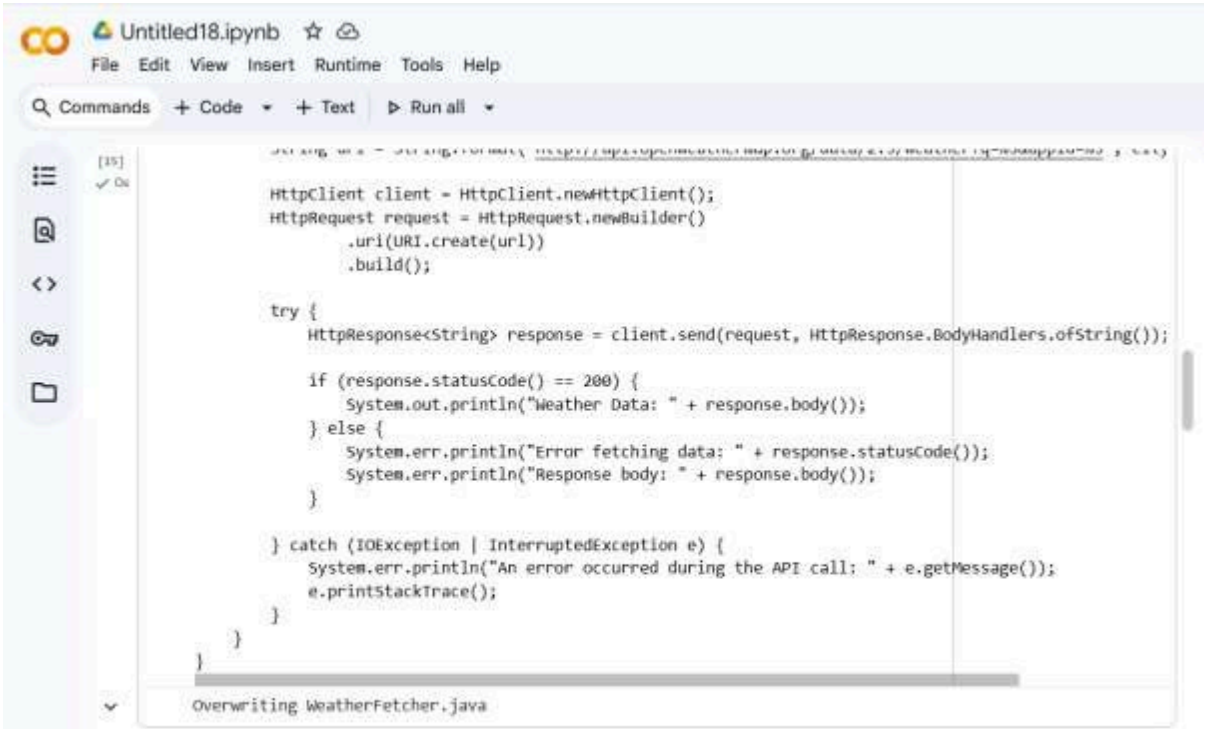
Printclearerrormessagesforeachcasewithouttheprogram crashing.

CODE:

```
importrequests


defget_weather():
```

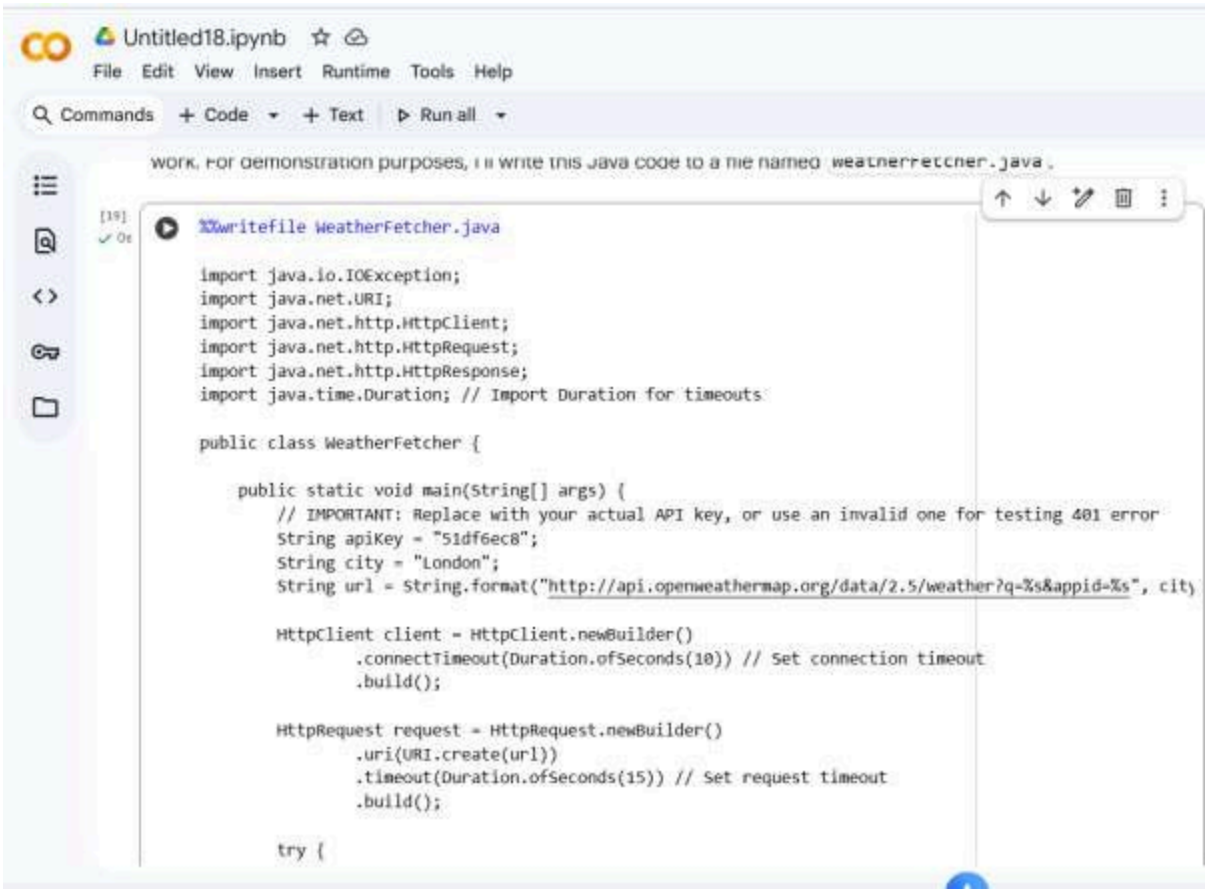
```
api_key="your_api_key_here"
city = "London"
url= f"http://api.openweathermap.org/data/2.5/weather?q={city}&appid={api_key} "

response=requests.get(url)
ifresponse.status_code==200:
    print("WeatherData:",response.json()
) else:
    print("Errorfetchingdata:",response.status_code)

get_weather()
```

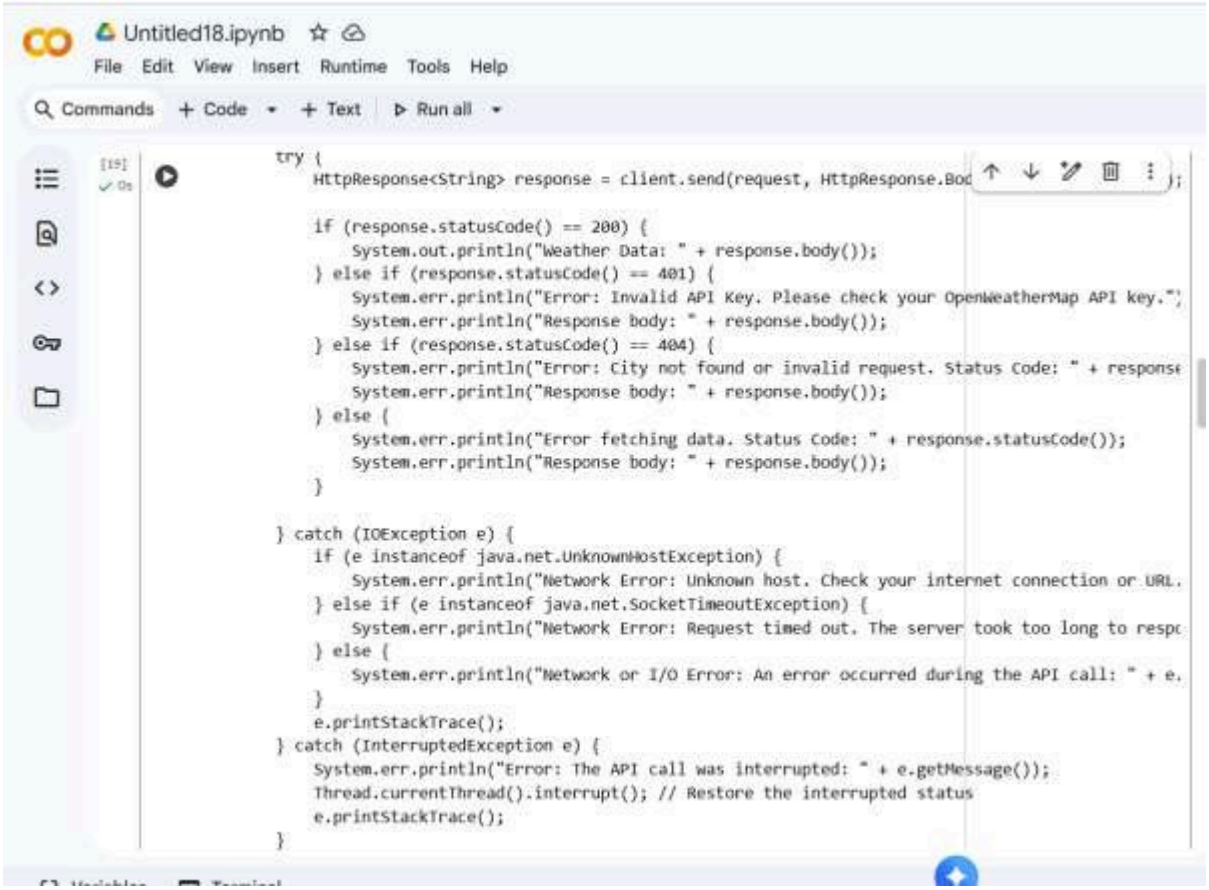




Now, let's compile and run the Java code.

Since Colab primarily supports Python, we'll compile and run the Java code using shell commands. Please ensure




Untitled18.ipynb
File Edit View Insert Runtime Tools Help
Commands + Code + Text Run all
try {
HttpResponse<String> response = client.send(request, HttpResponse.Bod
if (response.statusCode() == 200) {
System.out.println("Weather Data: " + response.body());
} else if (response.statusCode() == 401) {
System.err.println("Error: Invalid API Key. Please check your OpenWeatherMap API key.");
System.err.println("Response body: " + response.body());
} else if (response.statusCode() == 404) {
System.err.println("Error: City not found or invalid request. Status Code: " + response
System.err.println("Response body: " + response.body());
} else {
System.err.println("Error fetching data. Status Code: " + response.statusCode());
System.err.println("Response body: " + response.body());
}
} catch (IOException e) {
if (e instanceof java.net.UnknownHostException) {
System.err.println("Network Error: Unknown host. Check your internet connection or URL.
} else if (e instanceof java.net.SocketTimeoutException) {
System.err.println("Network Error: Request timed out. The server took too long to respo
} else {
System.err.println("Network or I/O Error: An error occurred during the API call: " + e.
}
e.printStackTrace();
} catch (InterruptedException e) {
System.err.println("Error: The API call was interrupted: " + e.getMessage());
Thread.currentThread().interrupt(); // Restore the interrupted status
e.printStackTrace();
}