

DAY-10

Task 2: Write a Python program that handles HTTP errors gracefully (e.g., 404 or 500 errors) when making API requests.

SOLUTION:

```
import requests
```

```
def fetch_url(url):
```

```
    try:
```

```
        response = requests.get(url)
```

```
        response.raise_for_status()    # Raise an
HTTPError for bad responses (4xx or 5xx)
```

```
        return response.content.decode('utf-8')    #
Assume content is text
```

```
    except requests.exceptions.HTTPError as http_err:
```

```
        print(f'HTTP error occurred: {http_err}')
```

```
    # Handle specific error codes
```

```
    if response.status_code == 404:

        print('Not Found: The resource you
requested could not be found.')

    elif response.status_code == 500:

        print('Internal Server Error: The server
encountered an unexpected condition.')

    # Handle other HTTP errors as needed

    else:

        print(f'Status Code:
{response.status_code}')

        print('An unexpected HTTP error
occurred.')

    except requests.exceptions.RequestException as
req_err:

        print(f'Request exception occurred: {req_err}')

    # Handle other request exceptions (connection
timeouts, etc.)
```

Example usage:

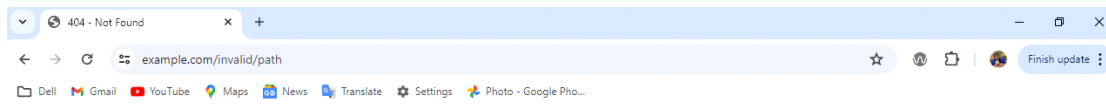
```
if __name__ == "__main__":
```

```
    url = 'https://example.com/invalid/path'    #
```

Example URL that might result in a 404 error

```
    fetch_url(url)
```

OUTPUT:



404 - Not Found