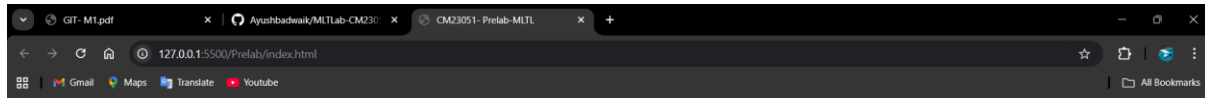


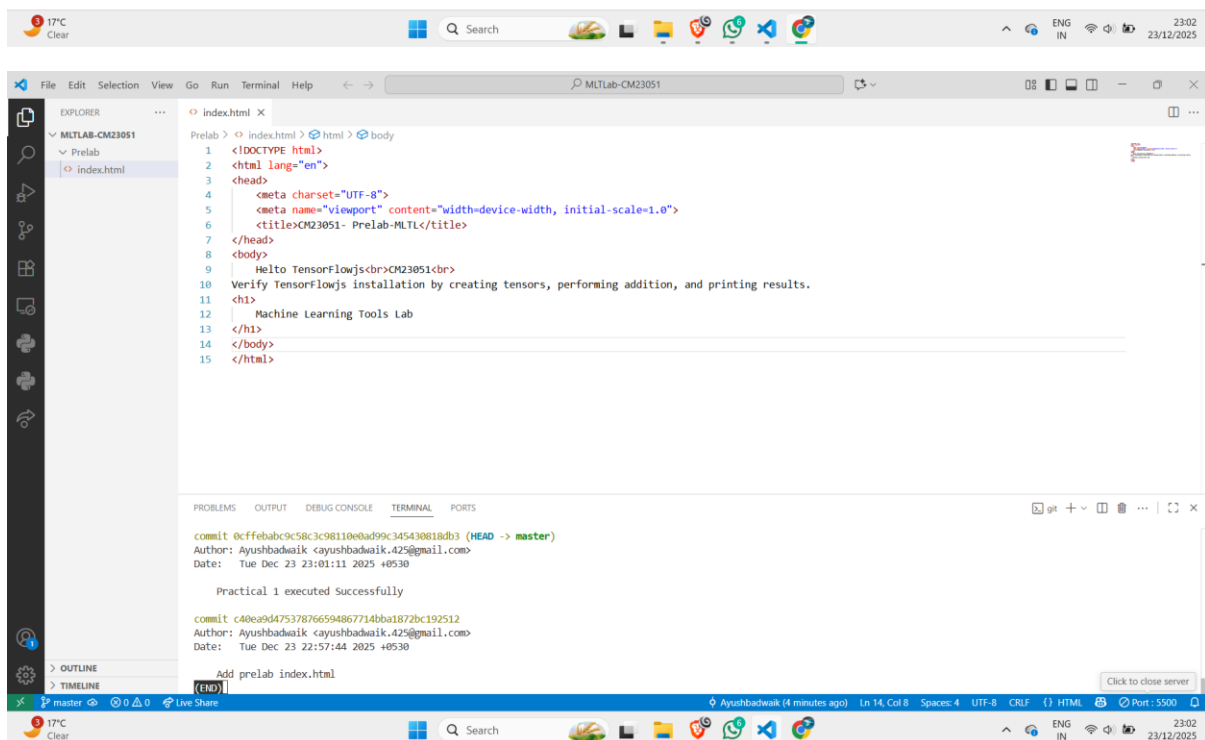
PRELAB OUTPUT

MACHINE LEARNING TOOLS LAB

AYUSH D. BADWAIK [CM23051]

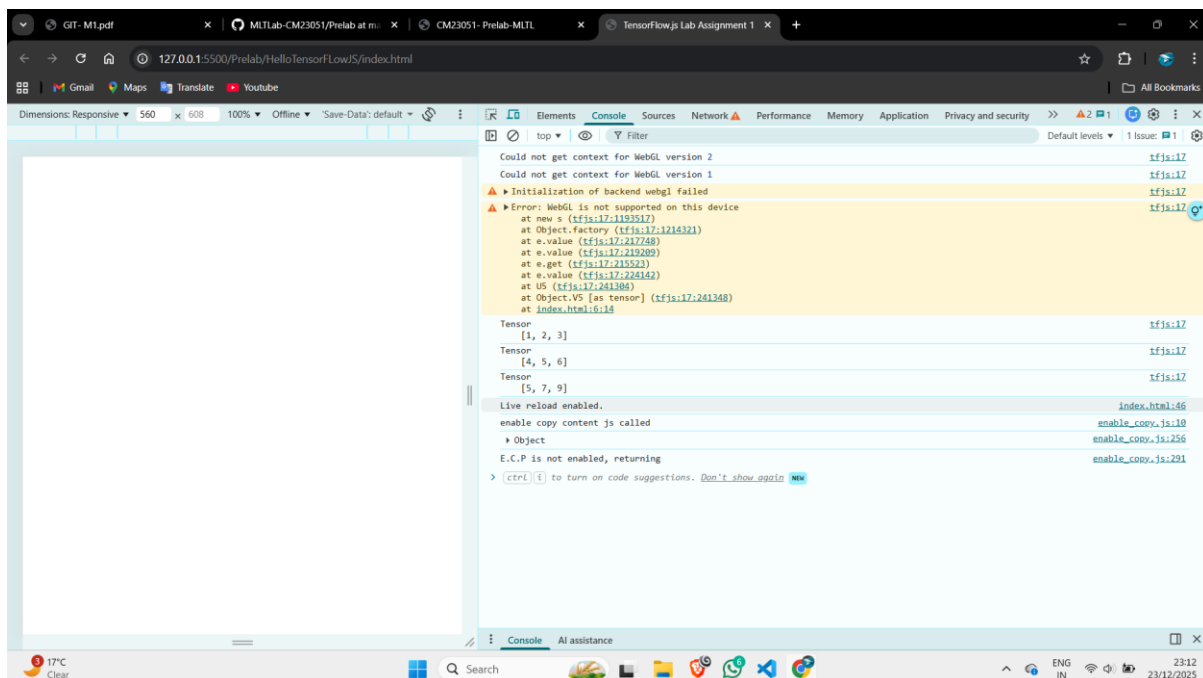


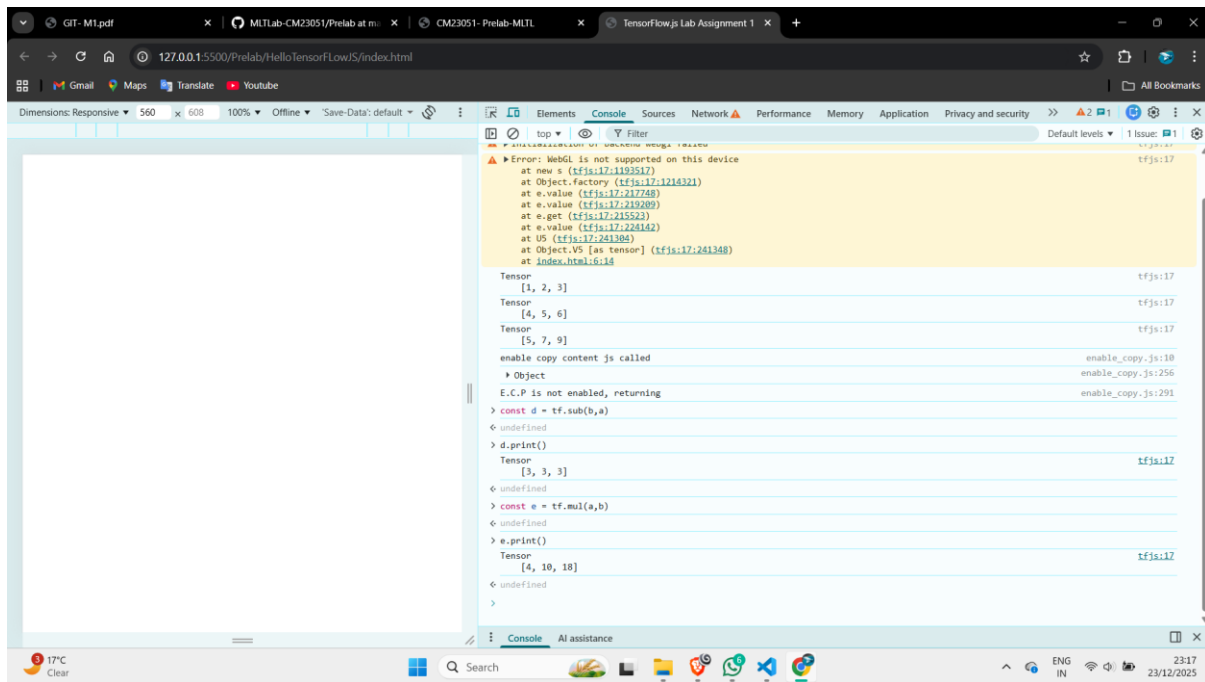
Machine Learning Tools Lab



PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

- PS D:\MLTLab-CM23051> git branch -M main
- PS D:\MLTLab-CM23051> git remote add origin https://github.com/Ayushbadwaik/MLTLab-CM23051.git
- PS D:\MLTLab-CM23051> git push -u origin main
Enumerating objects: 8, done.
Counting objects: 100% (8/8), done.
Delta compression using up to 12 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (8/8), 824 bytes | 824.00 KiB/s, done.
Total 8 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), done.
To https://github.com/Ayushbadwaik/MLTLab-CM23051.git
* [new branch] main -> main
branch 'main' set up to track 'origin/main'.
○ PS D:\MLTLab-CM23051> █





PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS D:\MLTLab-CM23051> git status

Changes to be committed:

(use "git restore --staged <file>..." to unstage)

new file: Prelab/HelloTensorFlowJS/index.html

- PS D:\MLTLab-CM23051> git commit -m "Assignment 1 successful Add HelloTensorFlowJS prelab index.html"

>>

[main b24ec09] Assignment 1 successful Add HelloTensorFlowJS prelab index.html

1 file changed, 19 insertions(+)

create mode 100644 Prelab/HelloTensorFlowJS/index.html

- PS D:\MLTLab-CM23051> git push -u origin main

Enumerating objects: 7, done.

Counting objects: 100% (7/7), done.

Delta compression using up to 12 threads

Compressing objects: 100% (3/3), done.

Writing objects: 100% (5/5), 622 bytes | 622.00 KiB/s, done.

Total 5 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)

To https://github.com/Ayushbadwaik/MLTLab-CM23051.git

0cffeba..b24ec09 main -> main

branch 'main' set up to track 'origin/main'.

- PS D:\MLTLab-CM23051>

The screenshot shows a web browser window with the URL `127.0.0.1:5500/Prelab/LabAssignment2/assign2.html`. The page title is "TensorFlow.js - Tensor Creation" by Arush D. Budwalk, CM23051. The page content says "Create tensors of different dimensions (scalar, vector, matrix) and print them." The console shows an error: "WebGL is not supported on this device" at `tfjs@latest:17:14321`. Below the error, a table displays tensor data:

Scalar Tensor:	assign2.html:19
Tensor	tfjs@latest:17
10	
Shape: <code>Array(0)</code>	assign2.html:21
Rank: 0	assign2.html:22
Vector Tensor:	assign2.html:28
Tensor	tfjs@latest:17
<code>[1, 2, 3, 4, 5]</code>	
Shape: <code>Array(1)</code>	assign2.html:30
Rank: 1	assign2.html:31
Matrix Tensor:	assign2.html:42
Tensor	tfjs@latest:17
<code>[[1, 2, 3],</code>	
<code>[4, 5, 6]]</code>	
Shape: <code>Array(2)</code>	assign2.html:44
Rank: 2	assign2.html:45
Live reload enabled.	assign2.html:75
enable copy content js called	enable_copy.js:10
Object	enable_copy.js:256
E.C.P is not enabled, returning	enable_copy.js:291

The screenshot shows a web browser window with the URL `127.0.0.1:5500/Prelab/LabAssignment3/assign3.html`. The page title is "Tensor Reshape and Flatten (TensorFlow.js)". The page content includes a "Concept Explanation" section, a "Difference Between reshape() and flatten()" section, and a "Live Demonstration" section.

Concept Explanation

reshape() and **flatten()** are tensor operations used to modify the shape of data without changing the total number of elements.

- reshape()** changes a tensor into a user-defined shape while keeping the same data.
- flatten()** converts a tensor of any dimension into a one-dimensional tensor.

Difference Between reshape() and flatten()

reshape():
- Output shape is user-defined
- Total elements must remain the same
- Used to reorganize tensor dimensions
flatten():
- Output is always 1-D
- No shape input required
- Used before feeding data into Dense layers

Live Demonstration

reshape():

- Output shape is user-defined
- Total elements must remain the same
- Used to reorganize tensor dimensions

flatten():

- Output is always 1-D
- No shape input required
- Used before feeding data into Dense layers

Live Demonstration

Run Tensor Operations

Original Tensor (2x3):

```
Tensor
[[1, 2, 3],
 [4, 5, 6]]
```

After reshape() → (3x2):

```
Tensor
[[1, 2],
 [3, 4],
 [5, 6]]
```

After Flatten() → (1-D):

```
Tensor
[1, 2, 3, 4, 5, 6]
```

github.com/Ayushbadwaik/MLTLab-CM23051/tree/main/Prelab

Ayushbadwaik / MLTLab-CM23051

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

Files

main

Go to file

Prelab

- HelloTensorFlowJS
 - index.html
- LabAssignment2
 - assign2.html
- LabAssignment3
 - assign3.html
 - index.html

MLTLab-CM23051 / Prelab

Ayushbadwaik 2nd & 3rd Assignment completed b30f405 · 3 minutes ago History

Name	Last commit message	Last commit date
..		
HelloTensorFlowJS	Assignment 1 successful Add HelloTensorFlowJS prelab index.html	21 minutes ago
LabAssignment2	2nd & 3rd Assignment completed	3 minutes ago
LabAssignment3	2nd & 3rd Assignment completed	3 minutes ago
index.html	Practical 1 executed Successfully	32 minutes ago