TEAM 77 FIRETRUCK

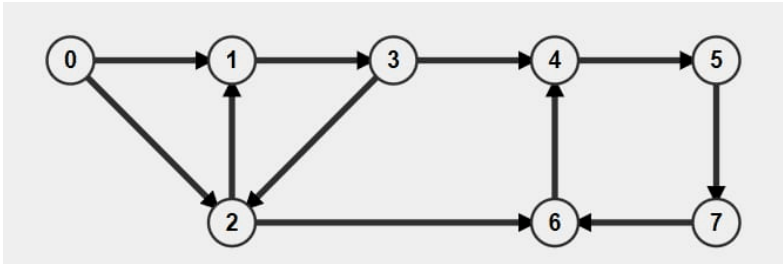
BVRIT HYDERABAD
RISHITA : 20WH1A0560 : CSE A
AMULYA : 20WH1A1294 : IT B
SANA : 21WH5A0406 : ECE A
RENUKA : 20WH1A0225 : EEE
ANUSHA : 20WH1A1238 : IT A
KEERTHANA : 20WH1A0420 : ECE A

27 OCTOBER

Introduction

- There is a fireplace and the fire station department need to reach the fireplace.

Approach

- 

- Note down the vertices
- Print all the possible paths from the starting point to the destination
- Create a graph for the given input

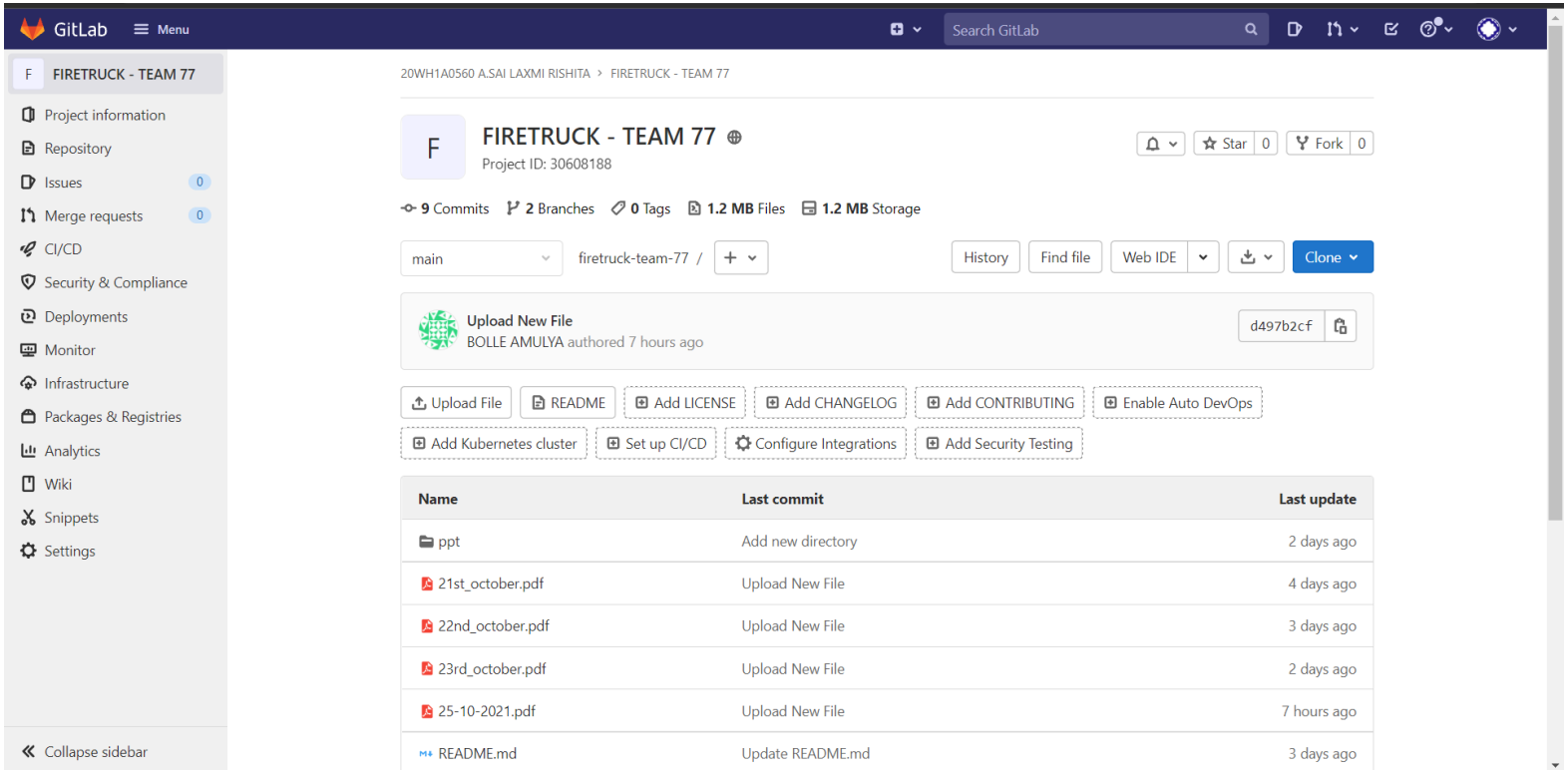
Learnings

- 1. LaTeX
- 2. Gitlab
- 3. Team work
- 4. Self Learning

Challenges

- Making our code generalized

GIT Repo


GitLab
Menu
Search GitLab
FIRETRUCK - TEAM 77
Project information
Repository
Issues 0
Merge requests 0
CI/CD
Security & Compliance
Deployments
Monitor
Infrastructure
Packages & Registries
Analytics
Wiki
Snippets
Settings
Collapse sidebar
20WH1A0560 A.SAI LAXMI RISHITA › FIRETRUCK - TEAM 77
FIRETRUCK - TEAM 77
Project ID: 30608188
Star 0
Fork 0
9 Commits
2 Branches
0 Tags
1.2 MB Files
1.2 MB Storage
main
firetruck-team-77 /
History
Find file
Web IDE
Clone
Upload New File
BOLLE AMULYA authored 7 hours ago
d497b2cf
Upload File
README
Add LICENSE
Add CHANGELOG
Add CONTRIBUTING
Enable Auto DevOps
Add Kubernetes cluster
Set up CI/CD
Configure Integrations
Add Security Testing
Name
Last commit
Last update
ppt
Add new directory
2 days ago
21st_october.pdf
Upload New File
4 days ago
22nd_october.pdf
Upload New File
3 days ago
23rd_october.pdf
Upload New File
2 days ago
25-10-2021.pdf
Upload New File
7 hours ago
README.md
Update README.md
3 days ago

Tech stack

- OS :Windows 10, Pycharm
- Language : Python
- LaTeX, Gitlab

Code

```python
#case 1 ie when street number 6 is nearest street to the fireplace
print("CASE 1:")
from collections import defaultdict
class Graph:
    def __init__(self,vertices):
        self.V= vertices
        self.graph = defaultdict(list)
    def addvertices(self,u,v):
        self.graph[u].append(v)
    def printAllPathsUtil(self, u, d, visited, path): #marking the current vertex as visited
        visited[u]= True
        path.append(u)
        if u==d:
            print(path)
        else:
            for i in self.graph[u]: #to remove the current vertex from the path amd we mark it as unvisite
                if visited[i]==False:
                    self.printAllPathsUtil(i, d, visited, path)
        path.pop()
        visited[u]= False
    def printAllPaths(self,s, d): # s is the start point and d is the destination point
        visited =[False]*(self.V)
        path = []
        self.printAllPathsUtil(s, d,visited, path) #to print all the paths
#creating a graph
g = Graph(8) #total 8 paths are given in the input
g.addvertices(1,2)
g.addvertices(1,3)
g.addvertices(3,4)
g.addvertices(3,5)
g.addvertices(4,6)
g.addvertices(5,6)
g.addvertices(2,3)
g.addvertices(3,4)
g.addvertices(0,0)
s = 1 ; d = 6
print ("These are the routes possible from the firestation to the streetcorner 6:")
g.printAllPaths(s, d)
```

Demo/ Screen Shots

- ```
  CASE 1:
  These are the routes possible from the firestation to the streetcorner
  [1, 2, 3, 4, 6]
  [1, 2, 3, 5, 6]
  [1, 2, 3, 4, 6]
  [1, 3, 4, 6]
  [1, 3, 5, 6]
  [1, 3, 4, 6]
  ```
  ```

THANK YOU!!