TASK REPORT – 2

Name: Samrudhi Shete

Task No	Task	Page No	Compled
1	How to login to GitLab using Golang	2	Complete
2	How to call GitLab API in Golang	5	Complete
3	how to create personal access token GitLab	8	Complete
4	how to create groups in Golang	9	Complete
5	how to create subgroup in Golang	9	Complete
6	how to create projects using Golang	9	Complete
7	how to generate excel sheet with your Golang project	13	Complete
8	How to build and run project through GitLab CICD	29	Complete
9	How to build go project using CICD on Gitlab	29	Complete
10	How to perform Git Local operations	36	Complete
11	OpenVPN and MoboXtreme	38	Complete

TASK 1: how to login to GitLab using Golang

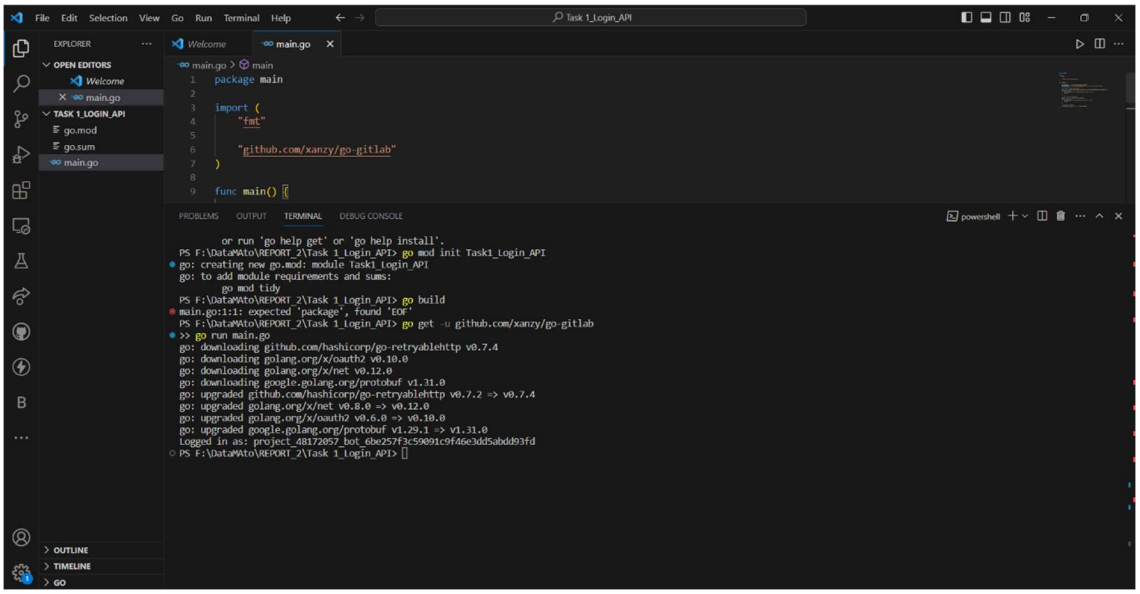
Code:

```
package main


import (
	"fmt"


	"github.com/xanzy/go-gitlab"
)


func main() {


	gitlabToken := "glpat-QYxpBHSxGp9xLEq8kB95"
	gitlabEndpoint := "https://gitlab.com" // Or your GitLab instance
URL


	// Create a GitLab API client
	git, err := gitlab.NewClient(gitlabToken,
gitlab.WithBaseURL(gitlabEndpoint))
	if err != nil {
		fmt.Println("Error creating GitLab client:", err)
		return
	}


	// Test the API connection
	user, _, err := git.Users.CurrentUser()
	if err != nil {
		fmt.Println("Error getting current user:", err)
		return
	}
```

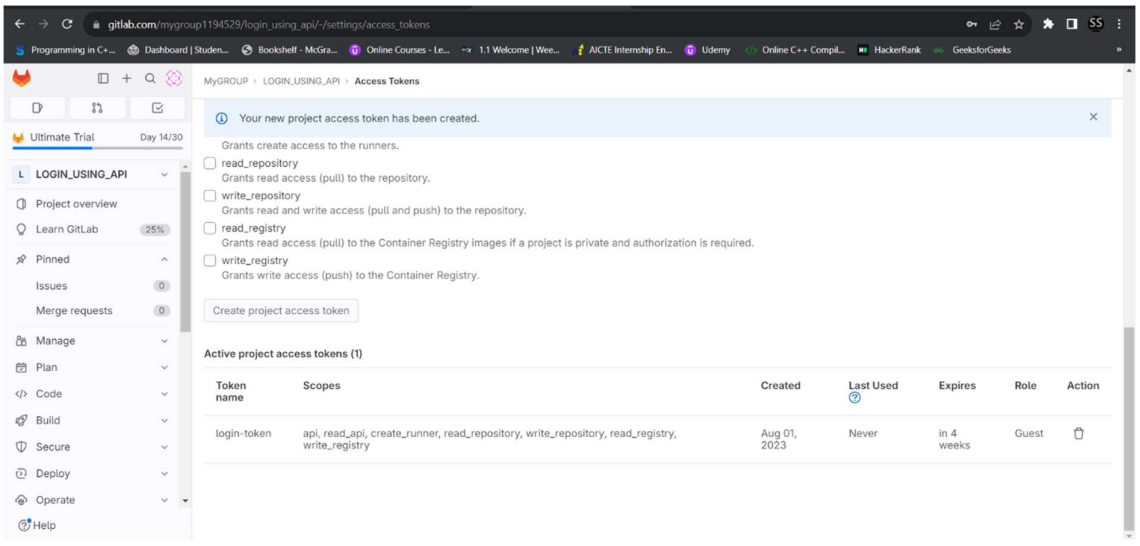
```
    // Print user details
    fmt.Println("Logged in as:", user.Username)
}
```

Output:

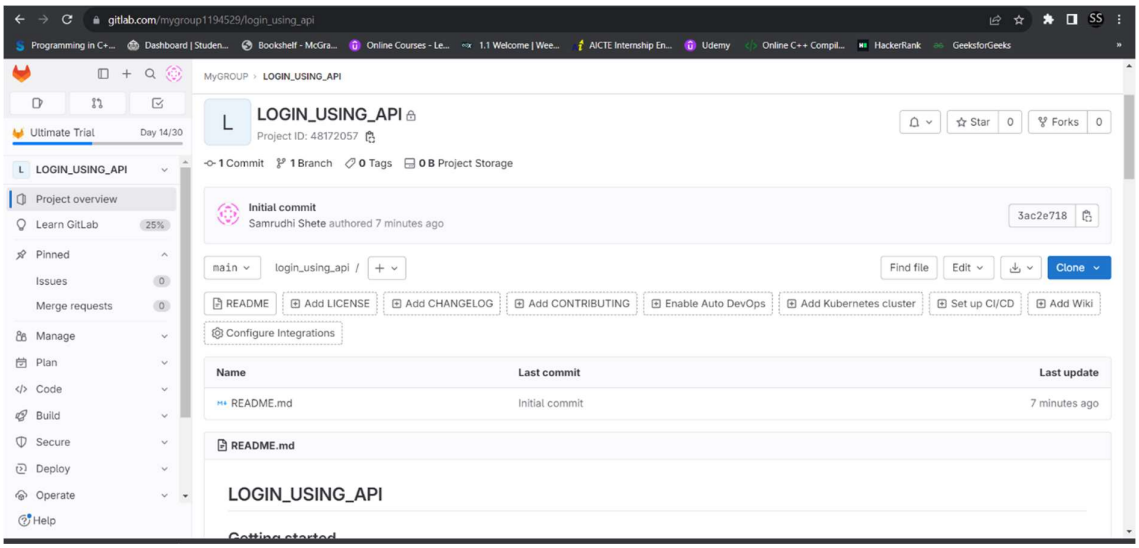
1.Output at Terminal



2.Personal Access Token used



3.Project



Issues: Completed Successfully

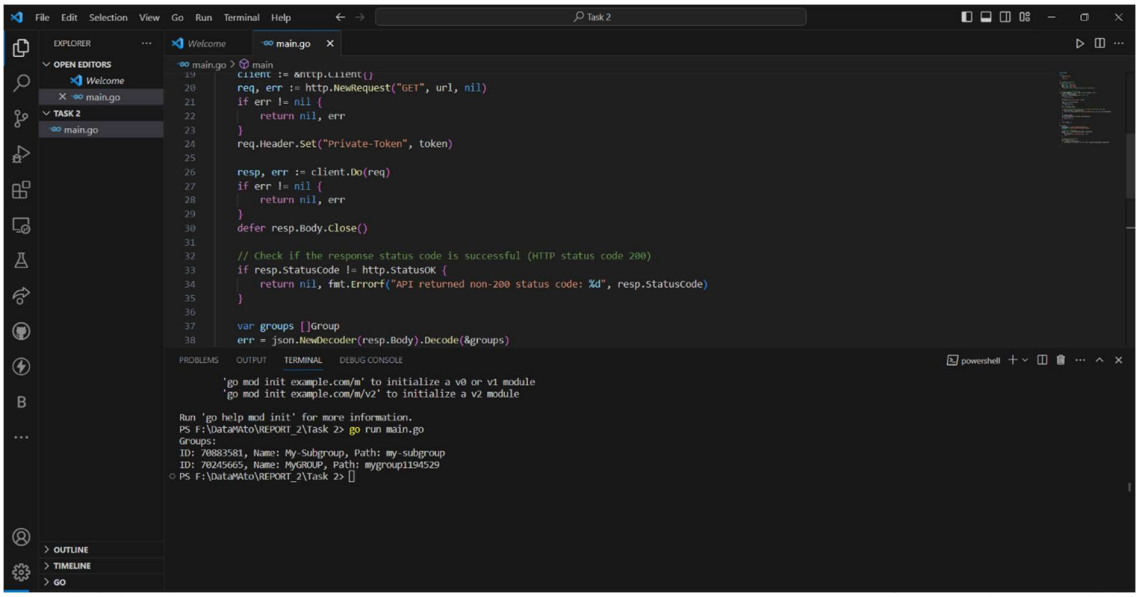
TASK 2: how to call GitLab API in Golang

Code:

```
package main

import (
	"encoding/json"
	"fmt"
	"net/http"
)

// GitLab Group structure
type Group struct {
	ID   int    `json:"id"`
	Name string `json:"name"`
	Path string `json:"path"`
	// Add more fields as needed based on your requirements
}

// Function to call the GitLab API
func callGitLabAPI(url string, token string) ([]Group, error) {
	client := &http.Client{}
	req, err := http.NewRequest("GET", url, nil)
	if err != nil {
		return nil, err
	}
	req.Header.Set("Private-Token", token)

	resp, err := client.Do(req)
	if err != nil {
		return nil, err
	}
	defer resp.Body.Close()

	// Check if the response status code is successful (HTTP status code 200)
```

```
	if resp.StatusCode != http.StatusOK {
		return nil, fmt.Errorf("API returned non-200 status code: %d",
resp.StatusCode)
	}

	var groups []Group
	err = json.NewDecoder(resp.Body).Decode(&groups)
	if err != nil {
		return nil, err
	}

	return groups, nil
}
func main() {
	gitLabToken := "glpat-5zUAG25QXqFz7Muej_Rf"
	gitLabURL := "https://gitlab.com/api/v4/groups"

	// Get a list of groups
	groups, err := callGitLabAPI(gitLabURL, gitLabToken)
	if err != nil {
		fmt.Println("Error getting groups:", err)
		return
	}
	// Print the list of groups
	fmt.Println("Groups:")
	for _, group := range groups {
		fmt.Printf("ID: %d, Name: %s, Path: %s\n", group.ID,
group.Name, group.Path)
	}

}
```

Output:



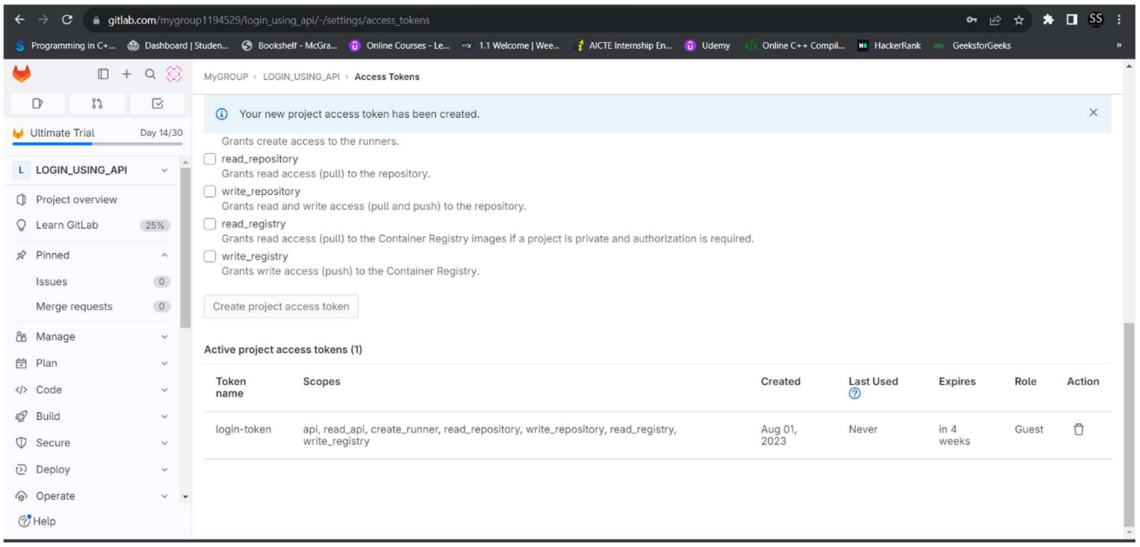
Issues: Completed Successfully

TASK 3: how to create personal access token GitLab

Code:

YOUR_PROJECT >> Settings >> Access Token >> Generate project access token

Output:



Issues: Completed Successfully

TASK 4 ,5 & 6: how to call groups and subgroup and projects in Gitlab using Golang API

Code:

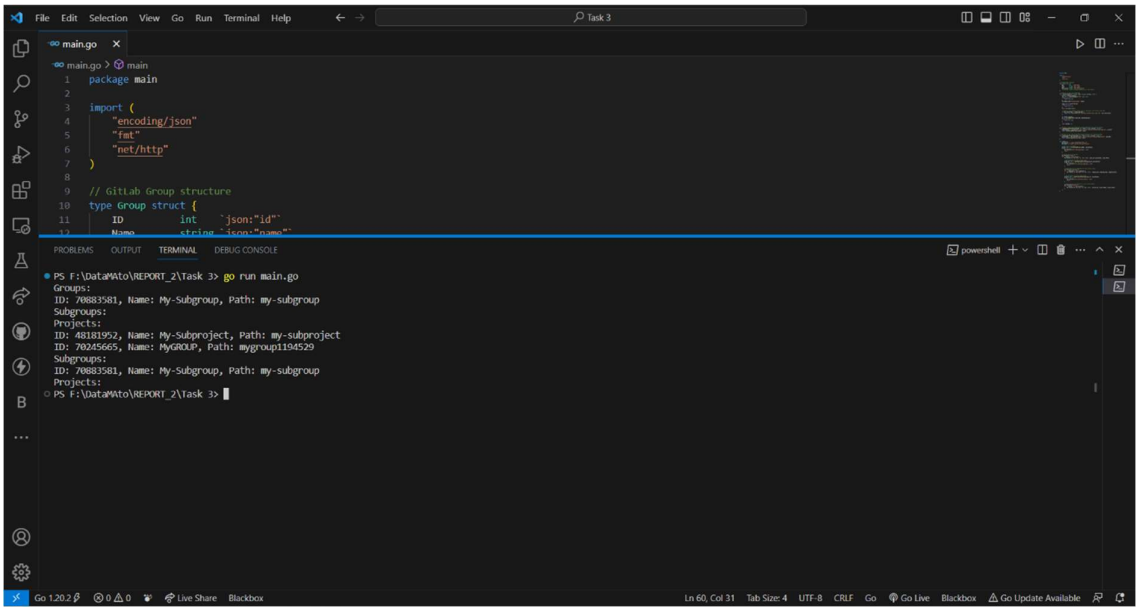
```
package main


import (
	"encoding/json"
	"fmt"
	"net/http"
)


// GitLab Group structure
type Group struct {
	ID          int    `json:"id"`
	Name        string `json:"name"`
	Path        string `json:"path"`
	Description string `json:"description"`
	// Add more fields as needed based on your requirements
}


// Function to call the GitLab API
func callGitLabAPI(url string, token string) ([]Group, error) {
	client := &http.Client{}
	req, err := http.NewRequest("GET", url, nil)
	if err != nil {
		return nil, err
	}
	req.Header.Set("Private-Token", token)


	resp, err := client.Do(req)
```

```
	if err != nil {
		return nil, err
	}
	defer resp.Body.Close()

	// Check if the response status code is successful (HTTP status code 200)
	if resp.StatusCode != http.StatusOK {
		return nil, fmt.Errorf("API returned non-200 status code: %d",
resp.StatusCode)
	}

	var groups []Group
	err = json.NewDecoder(resp.Body).Decode(&groups)
	if err != nil {
		return nil, err
	}

	return groups, nil
}

// Function to call the GitLab API and get subgroups for a specific group
func getSubgroupsForGroup(groupID int, token string) ([]Group, error) {
	subgroupsURL :=
fmt.Sprintf("https://gitlab.com/api/v4/groups/%d/subgroups", groupID)
	return callGitLabAPI(subgroupsURL, token)
}

// Function to call the GitLab API and get projects for a specific group
func getProjectsForGroup(groupID int, token string) ([]Group, error) {
	projectsURL := fmt.Sprintf("https://gitlab.com/api/v4/groups/%d/projects",
groupID)
	return callGitLabAPI(projectsURL, token)
}
```

```
func main() {
	gitLabToken := "glpat-5zUAG25QXqFz7Muej_Rf"
	gitLabURL := "https://gitlab.com/api/v4/groups"

	// Get a list of groups
	groups, err := callGitLabAPI(gitLabURL, gitLabToken)
	if err != nil {
		fmt.Println("Error getting groups:", err)
		return
	}

	// Print the list of groups
	fmt.Println("Groups:")
	for _, group := range groups {
		fmt.Printf("ID: %d, Name: %s, Path: %s\n", group.ID,
group.Name, group.Path)

		// Get subgroups for the current group
		subgroups, err := getSubgroupsForGroup(group.ID, gitLabToken)
		if err != nil {
			fmt.Println("Error getting subgroups:", err)
			return
		}
		// Print the list of subgroups for the current group
		fmt.Println("Subgroups:")
		for _, subgroup := range subgroups {
			fmt.Printf("ID: %d, Name: %s, Path: %s\n",
subgroup.ID, subgroup.Name, subgroup.Path)
		}
		// Get projects for the current group
		projects, err := getProjectsForGroup(group.ID, gitLabToken)
```

```
		if err != nil {
			fmt.Println("Error getting projects:", err)
			return
		}

		// Print the list of projects for the current group
		fmt.Println("Projects:")
		for _, project := range projects {
			fmt.Printf("ID: %d, Name: %s, Path: %s\n", project.ID,
project.Name, project.Path)
		}
	}
}
```

Output:



Issues: Completed Successfully

TASK 7: how to generate excel sheet with your Golang project

Code:

```
package main


//https://docs.gitlab.com/ee/integration/gitlab.html

import (
	"fmt"


	"math/rand"


	"strings"

	"time"


	"github.com/xuri/excelize/v2"

)


type Menu struct {
	itemNo    uint
	itemName  string
	itemPrice float64
}


var menu = []Menu{
	{1, "Adrakh Chai", 20},
	{2, "Filter Coffee", 25},
	{3, "Chhas", 35.50},
	{4, "Lassi", 30},
	{5, "Mango Lassi", 60},
	{6, "Kadi Chaawal", 50},
```

```
		{7, "Chhole Bhature", 45},
		{8, "Khasta Kachori", 30},
		{9, "Raj Kachori", 30},
		{10, "Veg. Sandwich", 20},
		{11, "Veg. Masala Maggi", 60.00},
		{12, "Samosa", 20},
		{13, "Cream Roll", 15},
}

var subTotalBill float64
var customerOrder = make(map[string]uint, 0)

func main() {
		var customerName string
		fmt.Println("Welcome here.....")
		fmt.Println("What is your first name?")
		fmt.Scan(&customerName)

		greet(customerName)
		orderItems()
		displayGeneratingBill()
		generateBill()
		modifyOrder()
		printFinalBill(customerName)
		sayTata(customerName)

		// Save order data to Excel file
		// Save order data to Excel file
		err := saveOrderDataToExcel(customerName)
		if err != nil {
```

```
                fmt.Println("Failed to save order data to Excel:", err)
                return
        }
        fmt.Println("Order data saved to Excel successfully!")
}
func greet(customerName string) {
        fmt.Printf("%52s %s\n", "Namaste ", customerName)
        fmt.Printf("%72s\n", "_/\\_ Welcome to Jaipur Bhojanalya! _/\\_ ")
        fmt.Println()
}

func sayTata(customerName string) {
        fmt.Println()
        fmt.Printf("%17s", " ")
        fmt.Printf("_/\\_ Thank you %v for visiting Jaipur Bhojanalya!
_/\\_\n", customerName)
        fmt.Printf("%55s\n", "We hope to see you again!")
        fmt.Printf("%51s\n", "Have a nice day! ")
}

func orderItems() {
        printMenu()
        var itemNumber uint
        var noOfPlates uint

        for {
                fmt.Println()
                fmt.Println("Enter '0' to exit.")
                fmt.Print("Enter the serial no. of the item to order: ")
```

```
		fmt.Scan(&itemNumber)
		if itemNumber == 0 {
			break
		}

		var choiceName string
		var itemPrice float64

		for index, item := range menu {
			if index+1 == int(itemNumber) {
				choiceName = item.itemName
				itemPrice = item.itemPrice
				break
			}
		}
		fmt.Printf("How many %v do you want: ", choiceName)
		fmt.Scan(&noOfPlates)
		if noOfPlates == 0 {
			continue
		} else {
			for index := range menu {
				if index+1 == int(itemNumber) {
					customerOrder[choiceName] +=
noOfPlates
					subTotalBill += itemPrice *
float64(noOfPlates)
					break
				}
			}
			fmt.Printf("\nYou just ordered %v %v which
amounts to ₹%v.\n", noOfPlates, choiceName,
itemPrice*float64(noOfPlates))
```

```
                orderTillNow()
            }
            fmt.Println()
        }
    }

    func orderTillNow() {
        fmt.Println("\nYour order till now: ")
        fmt.Printf("%s\n", strings.Repeat("-", 32))
        fmt.Printf(" %-12s %s\n", "Quantity", "Item")
        fmt.Printf("%s\n", strings.Repeat("-", 32))

        for i := range customerOrder {
            fmt.Printf(" %-12v %v\n", customerOrder[i], i)
        }

        fmt.Printf("%s\n", strings.Repeat("-", 32))
    }

    func modifyOrder() {
        for {
            var isOrderOK string
            fmt.Println("Do you want to change your order? [y/n]")
            fmt.Scan(&isOrderOK)
            if isOrderOK != "y" {
                return
            }
            var serialNo uint
            var modifyType uint
```

```
                fmt.Println("Please enter the respective number to
proceed:")
                fmt.Println("Press '1' to update item quantity.")
                fmt.Println("Press '2' to delete an item from the order list.")
                fmt.Println("Press '3' to add item(s) to the order list.")
                fmt.Scan(&modifyType)

                switch modifyType {
                case 1:
                        printMenu()
                        fmt.Println("Please enter the serial no. of the item
to be updated: ")
                        fmt.Scan(&serialNo)
                        updateQuantity(serialNo)

                case 2:
                        printMenu()
                        fmt.Println("Please enter the serial no. of the item
to be deleted: ")
                        fmt.Scan(&serialNo)
                        delFromOrder(serialNo)

                case 3:
                        insertIntoOrder()
                default:
                        return
                }
                displayGeneratingBill()
                generateBill()
        }
}
```

```
func updateQuantity(serialNo uint) {
	var newQuantity uint

	for _, targetItem := range menu {
		if serialNo == targetItem.itemNo {
			itemName := targetItem.itemName
			oldQuantity := customerOrder[itemName]

			fmt.Printf("Current quantity of %v is %v.\n",
itemName, oldQuantity)
			fmt.Printf("Enter the updated quantity of %v to be
ordered: \n", itemName)
			fmt.Scan(&newQuantity)

			if newQuantity == 0 {
				delFromOrder(serialNo)
				return
			}

			customerOrder[targetItem.itemName] =
newQuantity
			fmt.Printf("Updated the quantity of %v from %v
to %v.\n", itemName, oldQuantity, newQuantity)

			subTotalBill -= float64(oldQuantity) *
float64(targetItem.itemPrice)
			subTotalBill += float64(newQuantity) *
float64(targetItem.itemPrice)

			break
		}
	}
```

```
}


func delFromOrder(serialNo uint) {
	for _, targetItem := range menu {
		if serialNo == targetItem.itemNo {
			itemName := targetItem.itemName
			subTotalBill -=
float64(customerOrder[itemName]) * float64(targetItem.itemPrice)
			delete(customerOrder, itemName)
			fmt.Printf("%v removed from the order list.\n",
itemName)
			break
		}
	}
}


func insertIntoOrder() {
	orderItems()
}


func printMenu() {
	fmt.Printf("%15s\n", "Menu")
	fmt.Printf("%s\n", strings.Repeat("-", 35))
	fmt.Printf("%-7s %6s    %12s\n", "S.No.", "Price", "Item Name")
	fmt.Printf("%s\n", strings.Repeat("-", 35))


	for _, element := range menu {
		fmt.Printf(" %-7d %.2f    %-4s\n", element.itemNo,
element.itemPrice, element.itemName)
	}
```

```
	fmt.Printf("%s\n", strings.Repeat("-", 35))

	fmt.Println()

}


// just beautifying my code :P

func displayGeneratingBill() {

	fmt.Println()

	billDisplayText := "************************************
Generating Bill ************************************"

	for _, element := range billDisplayText {

		fmt.Printf("%c", element) // if you use "%v" instead of
"%c" then convert element into string, as shown in the comment below

		// fmt.Print("%v", string(element))

		time.Sleep(time.Millisecond * 15)

	}

}


// prints item name, price, quantity and total price and sub total amount.

func generateBill() {



	fmt.Println()

	fmt.Printf("+%s+\n", strings.Repeat("-", 90))

	fmt.Printf(" %-30s %-20s %-20s %-20s\n", "Item Name",
"Price(₹)", "Quantity", "Total Price(₹)")

	fmt.Printf("+%s+\n", strings.Repeat("-", 90))



	//prints the details of the food item that you've ordered.

	printOrderData()



	fmt.Printf("+%s+\n", strings.Repeat("-", 90))
```

```
	//print sub total amount in a cool way!
	fmt.Printf("%47s", " ")
	for _, element := range "Sub Total(excluding GST): ₹" {
		fmt.Printf("%c", element)
		time.Sleep(time.Millisecond * 50)
	}
	fmt.Printf("%.2f\n", subTotalBill)

}

// prints the data of the items that you ordered.
func printOrderData() {
	for key := range customerOrder {
		//key -> it is the key values
		for _, element := range menu {
			if key == element.itemName {
				//customerOrder[key] -> will provide the
no. of plates of that item

	//float64(customerOrder[key])*element.itemPrice -> this will result
in the cost of each item
				totalCostOfItem :=
float64(customerOrder[key]) * element.itemPrice
				fmt.Printf(" %-30s %-20.2f %-20v %-
20.2f\n", key, element.itemPrice, customerOrder[key], totalCostOfItem)
			}
		}
	}
	fmt.Println()
}

func printFinalBill(customerName string) {
```

```
	for _, element := range "Here is your final bill:-" {
		fmt.Printf("%c", element)
		time.Sleep(time.Millisecond * 50)
	}
	fmt.Println()

	fmt.Printf("\n%52s\n", "JAIPUR BHOJANALYA")
	time.Sleep(time.Millisecond * 200)
	fmt.Printf("%s\n", strings.Repeat("*", 91))
	time.Sleep(time.Millisecond * 200)
	fmt.Printf("%86s\n", "Bhawani Singh Road, First Floor, Jaipur
Bhojanalya, Jaipur, Jaipur 302005, Bharat")
	time.Sleep(time.Millisecond * 200)
	fmt.Printf("%50s\n", "Tel: 92145623XX")
	fmt.Printf("%60s\n\n", "Email: jaipur.bhojanalaya@gmail.com")
	time.Sleep(time.Millisecond * 200)
	fmt.Printf("%s", strings.Repeat("-", 42))
	fmt.Printf("%s", "INVOICE")
	fmt.Printf("%s\n", strings.Repeat("-", 42))
	time.Sleep(time.Millisecond * 200)

	rand.Seed(time.Now().Unix()) //necessary to produce random
integers
	fmt.Printf(" Ticket No: %d\n", rand.Intn(550)+1)

	fmt.Printf(" Date: %v\n", time.Now().Local().Format("06-Jan-02"))
//display date
	fmt.Printf(" Time: %v", time.Now().Local().Format("3:4 PM"))
//display time
	time.Sleep(time.Millisecond * 200)

	generateBill() //prints details of the bill,like, item name, price,
```

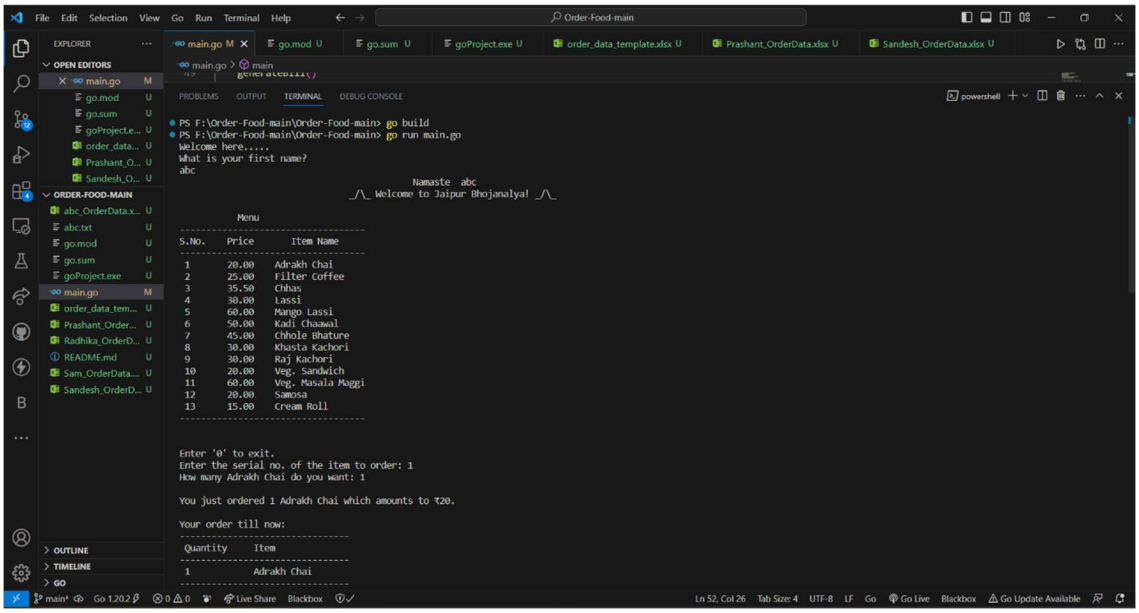
quantity and total price and sub total amount.

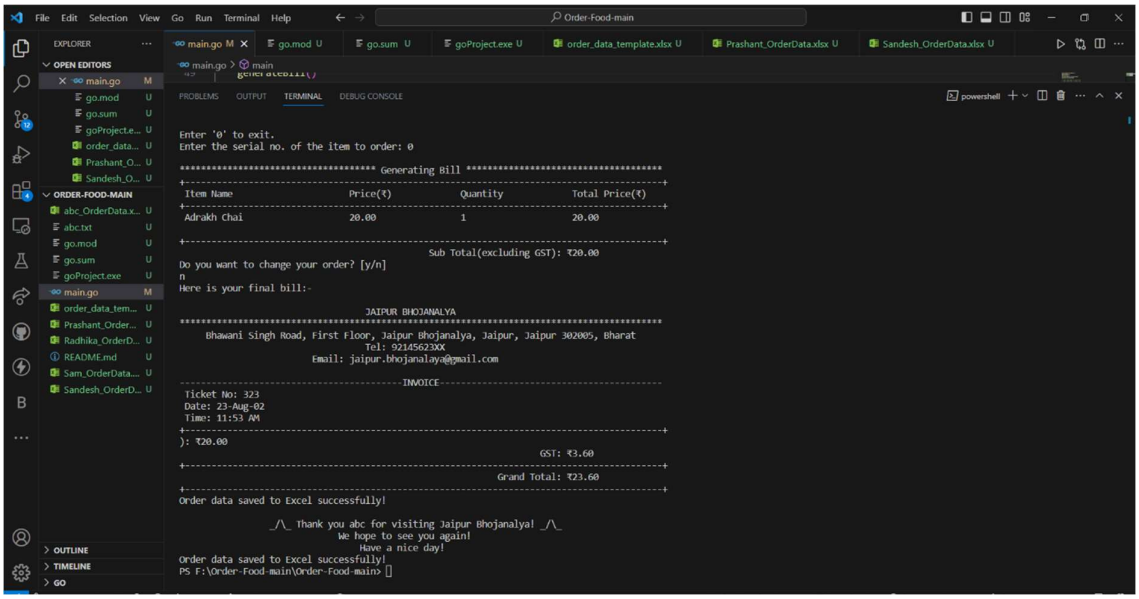
```
	tax := 18 * subTotalBill / (100)
	grandTotal := subTotalBill + tax

	time.Sleep(time.Millisecond * 200)
	fmt.Printf("%71s: ₹%.2f\n", "GST", tax) //display tax amount
	fmt.Printf("+%s+\n", strings.Repeat("-", 90))
	time.Sleep(time.Millisecond * 200)
	fmt.Printf("%71s: ₹%.2f\n", "Grand Total", grandTotal) //display
final bill
	fmt.Printf("+%s+\n", strings.Repeat("-", 90))

	err := saveOrderDataToExcel(customerName)
	if err != nil {
		fmt.Println("Failed to save order data to Excel:", err)
		return
	}
	fmt.Println("Order data saved to Excel successfully!")
}

// Funstion to call the gitlab api
// here the function is for to make the api call and retrieve data
// Personal access token
// glpat-7yDJQhx6ngTqdusLzksK

////excel file

// Function to save order data to Excel file
func saveOrderDataToExcel(customerName string) error {
```

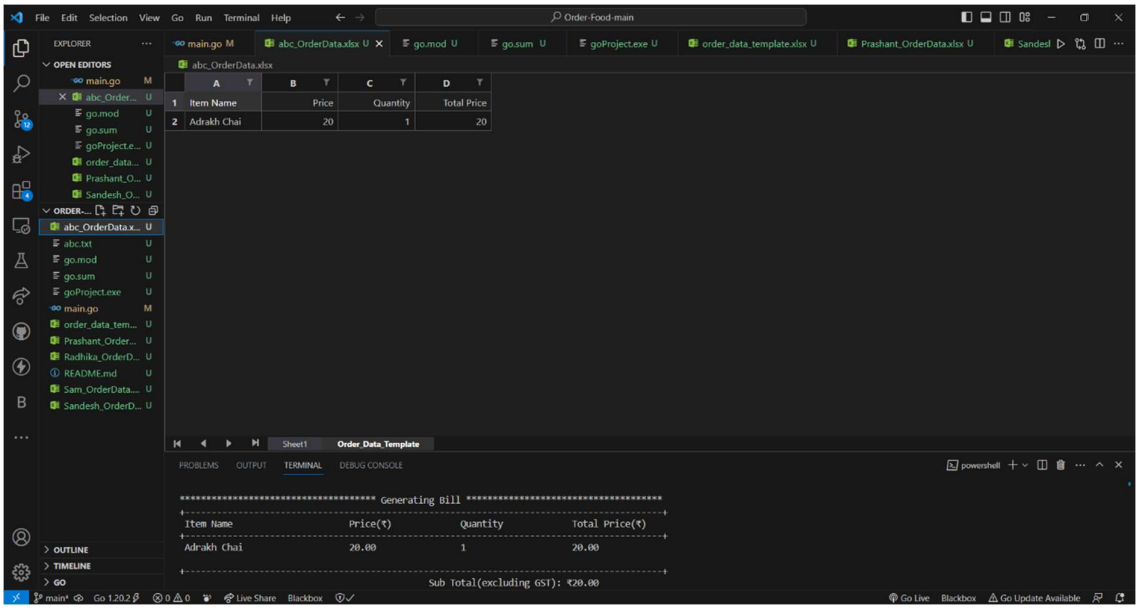
```
	file := excelize.NewFile()
	sheetName := "Order_Data_Template"

	// Create a new sheet
	index, err := file.NewSheet(sheetName)
	if err != nil {
		return err
	}

	// Set the column headers
	file.SetCellValue(sheetName, "A1", "Item Name")
	file.SetCellValue(sheetName, "B1", "Price")
	file.SetCellValue(sheetName, "C1", "Quantity")
	file.SetCellValue(sheetName, "D1", "Total Price")

	// Write the order data to the sheet
	row := 2
	for key, value := range customerOrder {
		itemName := key
		quantity := value
		itemPrice := getItemPrice(itemName)
		totalPrice := itemPrice * float64(quantity)

		file.SetCellValue(sheetName, fmt.Sprintf("A%d", row),
itemName)
		file.SetCellValue(sheetName, fmt.Sprintf("B%d", row),
itemPrice)
		file.SetCellValue(sheetName, fmt.Sprintf("C%d", row),
quantity)
		file.SetCellValue(sheetName, fmt.Sprintf("D%d", row),
totalPrice)
```

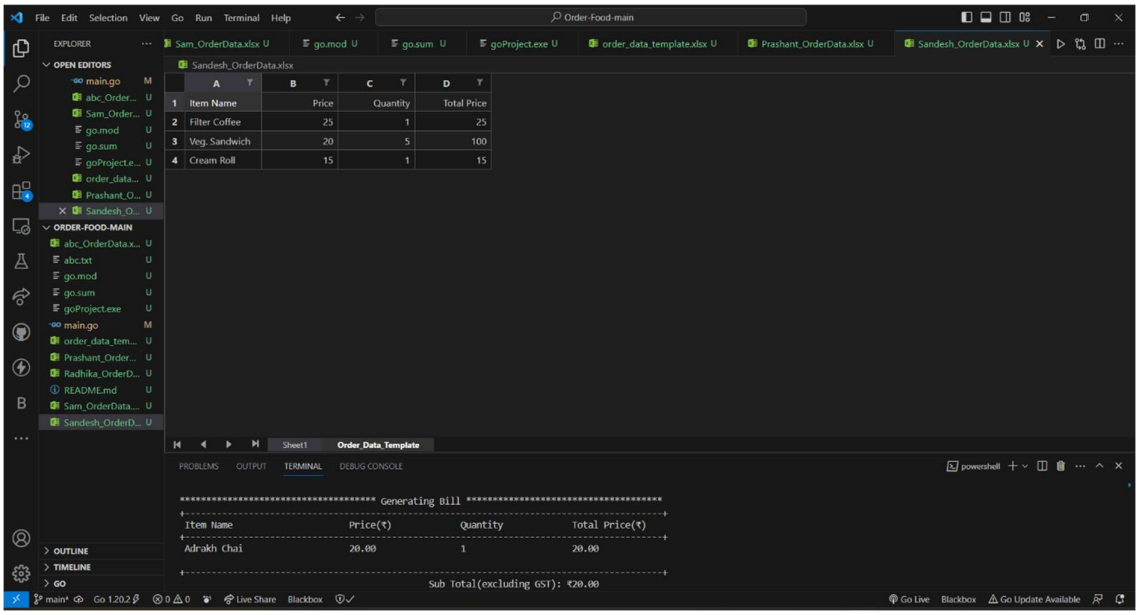
```
		row++
	}

	// Set the active sheet
	file.SetActiveSheet(index)

	// Save the Excel file
	err = file.SaveAs(customerName + "_OrderData.xlsx")
	if err != nil {
		return err
	}

	return nil
}

func getItemPrice(itemName string) float64 {
	for _, item := range menu {
		if item.itemName == itemName {
			return item.itemPrice
		}
	}
	panic(fmt.Sprintf("Item '%s' not found in the menu.", itemName))
}
```

Output:


PS F:\Order-Food-main\Order-Food-main> go build
PS F:\Order-Food-main\Order-Food-main> go run main.go
Welcome here.....
What is your first name?
abc
Namaste abc
_/_ Welcome to Jaipur Bhojanalya! _/_
Menu
S.No. Price Item Name
1 20.00 Adrakh Chai
2 25.00 Filter Coffee
3 35.50 Chhas
4 30.00 Lassi
5 60.00 Mango Lassi
6 50.00 Kadi Chaawal
7 45.00 Chhole Bhature
8 30.00 Khasta Kachori
9 30.00 Raj Kachori
10 20.00 Veg. Sandwich
11 60.00 Veg. Masala Maggi
12 20.00 Samosa
13 15.00 Cream Roll
Enter '0' to exit.
Enter the serial no. of the item to order: 1
How many Adrakh Chai do you want: 1
You just ordered 1 Adrakh Chai which amounts to ₹20.
Your order till now:
Quantity Item
1 Adrakh Chai


Enter '0' to exit.
Enter the serial no. of the item to order: 0
Generating Bill
Item Name Price(₹) Quantity Total Price(₹)
Adrakh Chai 20.00 1 20.00
Sub Total(excluding GST): ₹20.00
Do you want to change your order? [y/n]
n
Here is your final bill:-
JAIPUR BHOJANALYA
Bhawani Singh Road, First Floor, Jaipur Bhojanalya, Jaipur, Jaipur 302005, Bharat
Tel: 92145623XX
Email: jaipur.bhojanalaya@gmail.com
INVOICE
Ticket No: 323
Date: 23-Aug-02
Time: 11:53 AM
): ₹20.00
GST: ₹3.60
Grand Total: ₹23.60
Order data saved to Excel successfully!
_/_ Thank you abc for visiting Jaipur Bhojanalya! _/_
We hope to see you again!
Have a nice day!
Order data saved to Excel successfully!
PS F:\Order-Food-main\Order-Food-main>





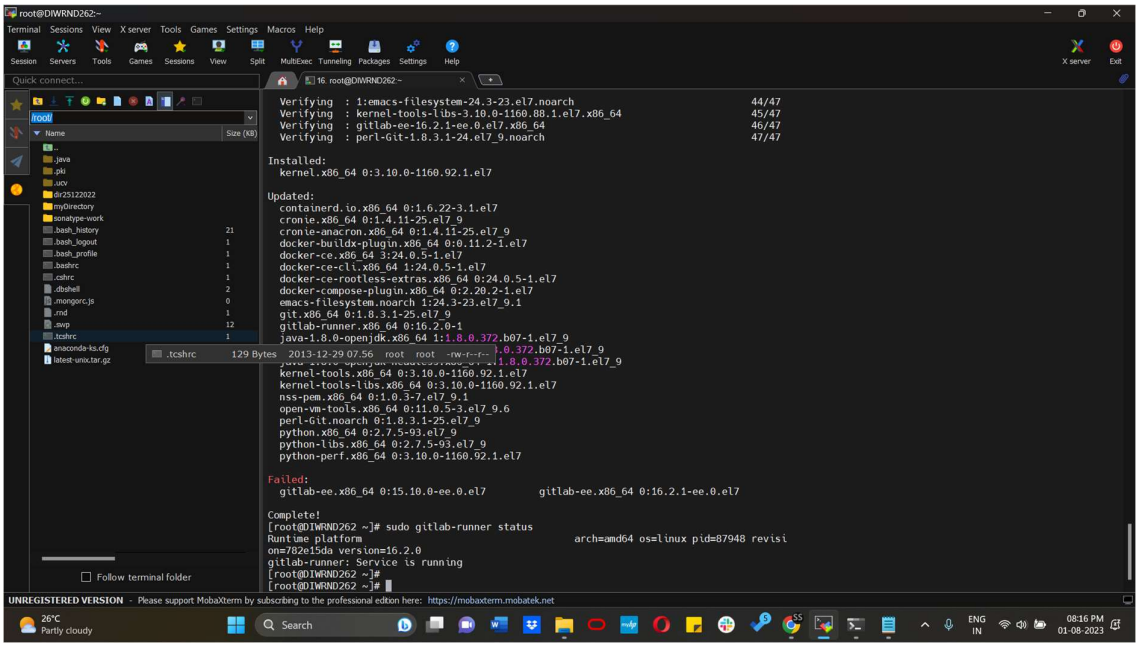
Issues: Completed Successfully

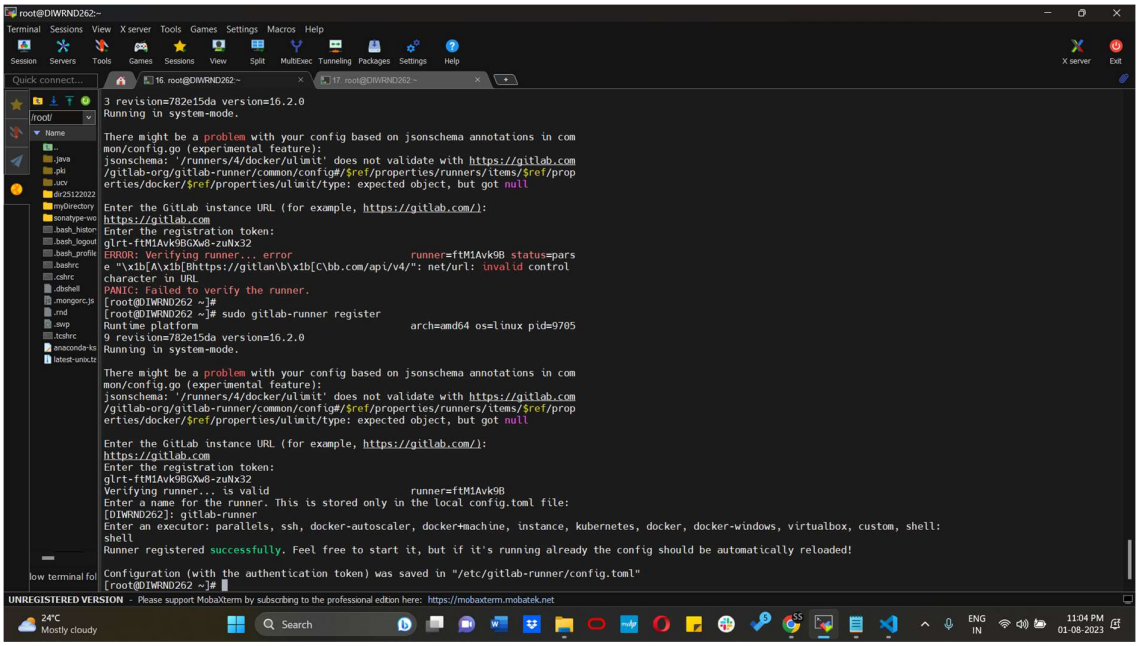
TASK 8 & 9: how to build run go project using CICD

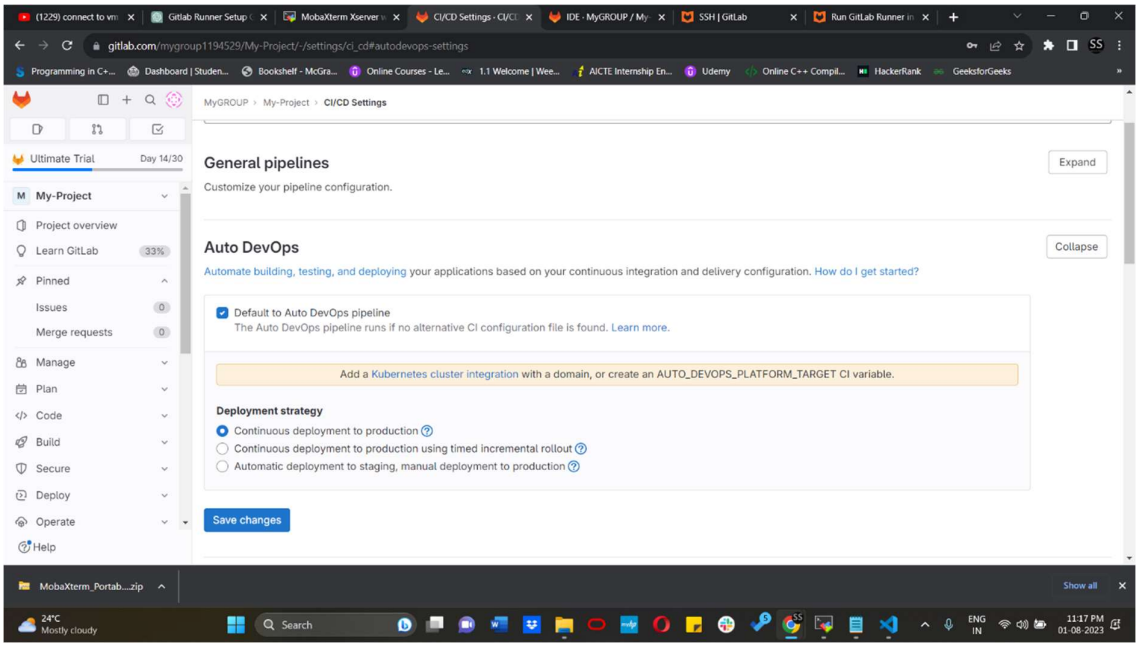
Code:

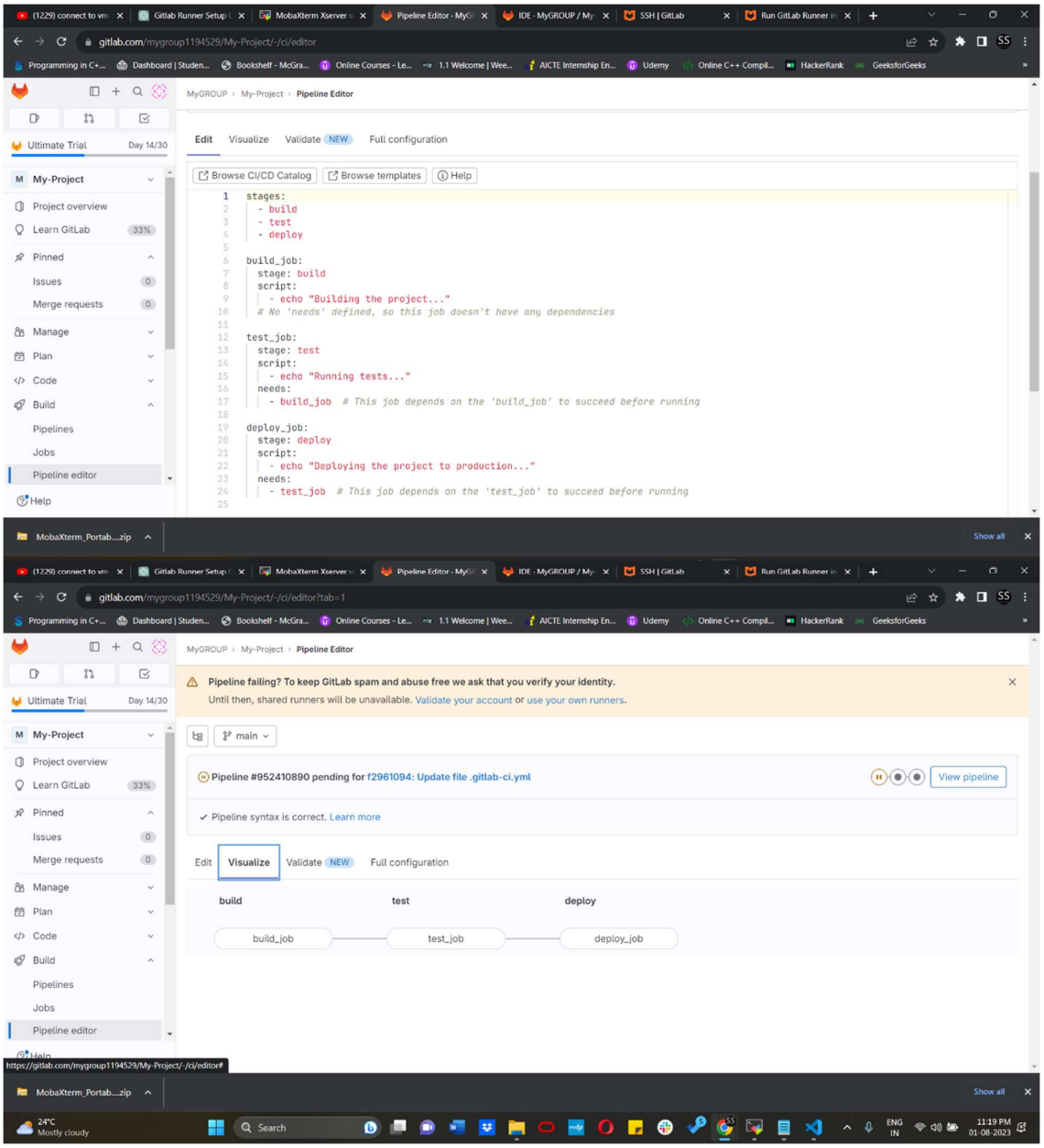
```
stages:
  - build
  - test
  - deploy

build_job:
  stage: build
  script:
    - echo "Building the project..."
  # No 'needs' defined, so this job doesn't have any dependencies

test_job:
  stage: test
  script:
    - echo "Running tests..."
  needs:
    - build_job  # This job depends on the 'build_job' to succeed before
running

deploy_job:
  stage: deploy
  script:
    - echo "Deploying the project to production..."
  needs:
    - test_job  # This job depends on the 'test_job' to succeed before
running
```

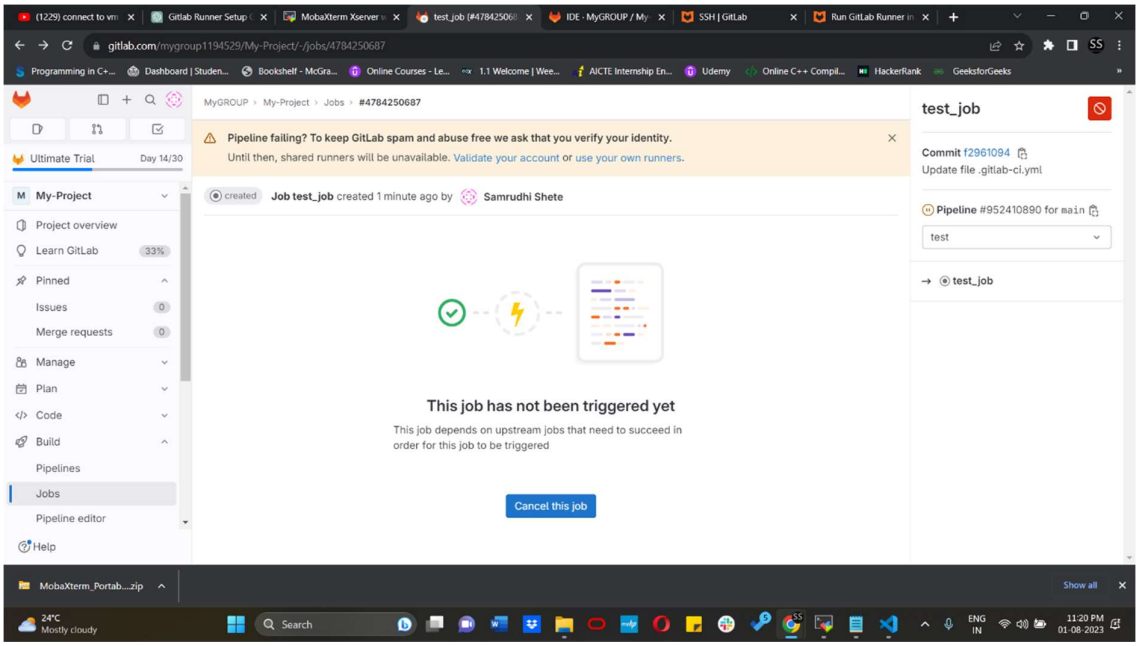
Output:

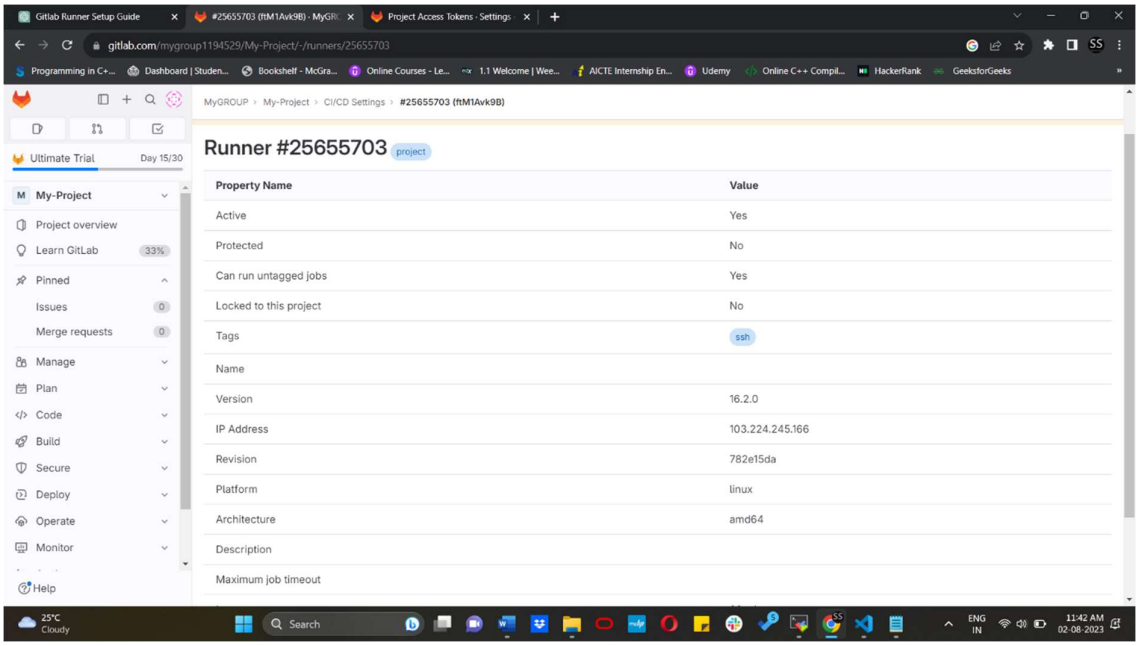


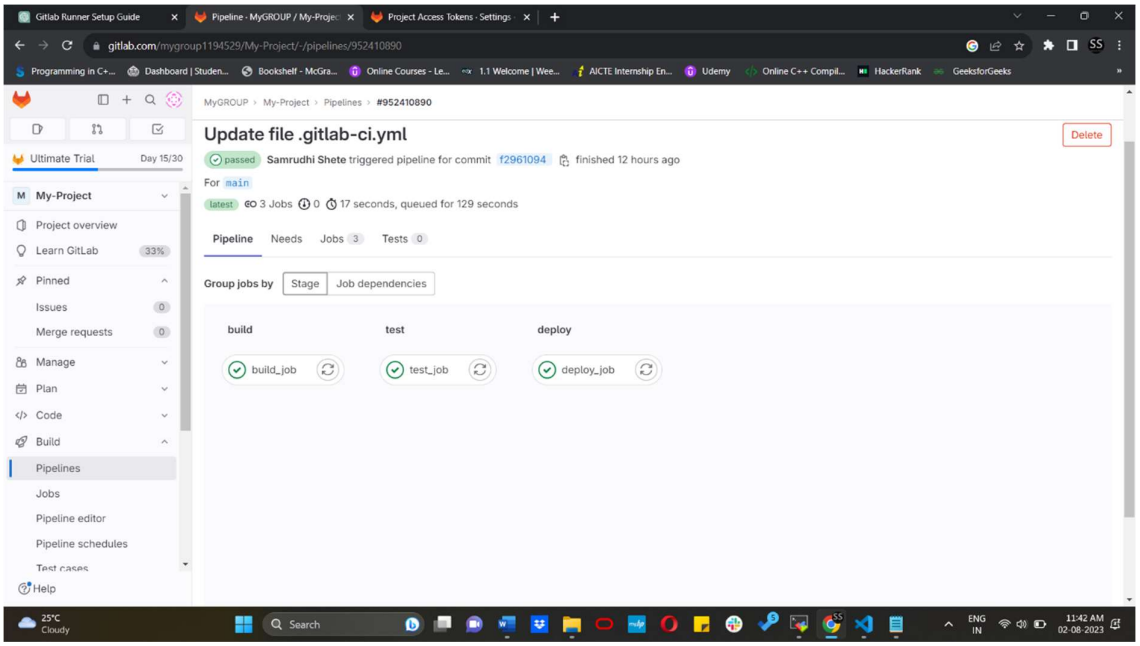

root@DIWRND262:~
3 revision=782e15da version=16.2.0
Running in system-mode.
There might be a problem with your config based on jsonschema annotations in com
mon/config.go (experimental feature):
jsonschema: '/runners/4/docker/ulimit' does not validate with https://gitlab.com
/gitlab-org/gitlab-runner/common/config#/$ref/properties/runners/items/$ref/prop
erties/docker/$ref/properties/ulimit/type: expected object, but got null
Enter the GitLab instance URL (for example, https://gitlab.com/):
https://gitlab.com
Enter the registration token:
glrt-ftM1Avk9BGXw8-zuNx32
ERROR: Verifying runner... error runner=ftM1Avk9B status=pars
e "\x1b[A\x1b[8https://gitlan\b\x1b[C\bb.com/api/v4/": net/url: invalid control
character in URL
PANIC: Failed to verify the runner.
[root@DIWRND262 ~]#
[root@DIWRND262 ~]# sudo gitlab-runner register
Runtime platform arch=amd64 os=linux pid=9705
9 revision=782e15da version=16.2.0
Running in system-mode.
There might be a problem with your config based on jsonschema annotations in com
mon/config.go (experimental feature):
jsonschema: '/runners/4/docker/ulimit' does not validate with https://gitlab.com
/gitlab-org/gitlab-runner/common/config#/$ref/properties/runners/items/$ref/prop
erties/docker/$ref/properties/ulimit/type: expected object, but got null
Enter the GitLab instance URL (for example, https://gitlab.com/):
https://gitlab.com
Enter the registration token:
glrt-ftM1Avk9BGXw8-zuNx32
Verifying runner... is valid runner=ftM1Avk9B
Enter a name for the runner. This is stored only in the local config.toml file:
[DIWRND262]: gitlab-runner
Enter an executor: parallels, ssh, docker-autoscaler, docker+machine, instance, kubernetes, docker, docker-windows, virtualbox, custom, shell:
shell
Runner registered successfully. Feel free to start it, but if it's running already the config should be automatically reloaded!
Configuration (with the authentication token) was saved in "/etc/gitlab-runner/config.toml"
[root@DIWRND262 ~]#
UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: https://mobaxterm.mobatek.net

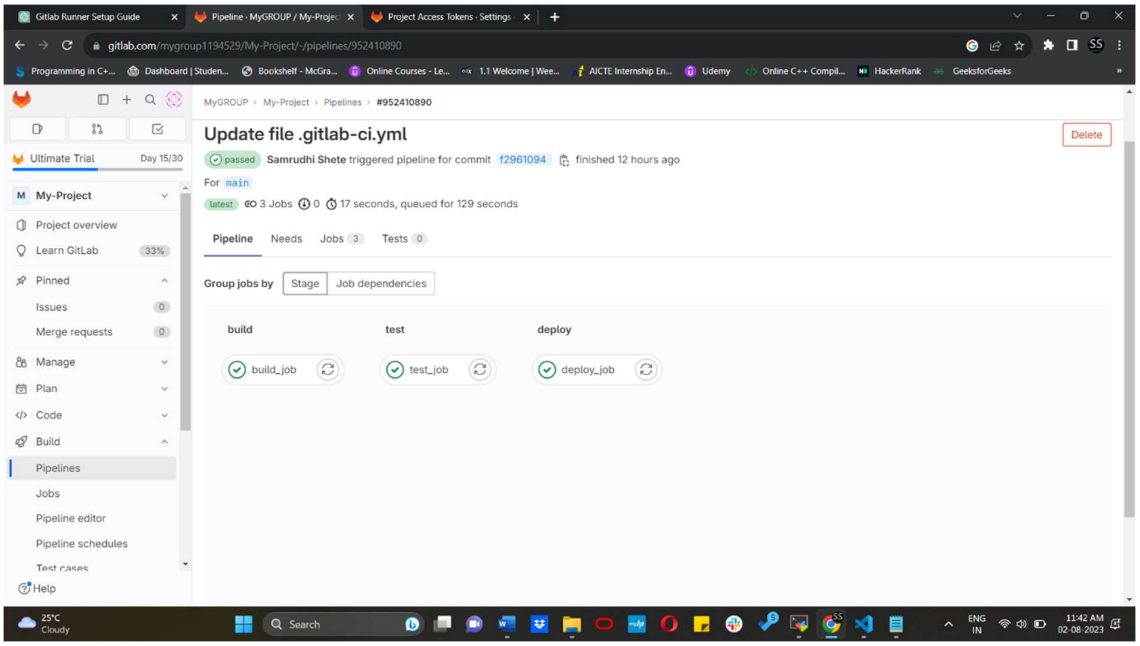

gitlab.com/mygroup1194529/My-Project/-/settings/ci_cd#autodevops-settings
MyGROUP › My-Project › CI/CD Settings
Ultimate Trial Day 14/30
My-Project
Project overview
Learn GitLab 33%
Pinned
Issues 0
Merge requests 0
Manage
Plan
Code
Build
Secure
Deploy
Operate
Help
General pipelines
Expand
Customize your pipeline configuration.
Auto DevOps
Collapse
Automate building, testing, and deploying your applications based on your continuous integration and delivery configuration. How do I get started?
Default to Auto DevOps pipeline
The Auto DevOps pipeline runs if no alternative CI configuration file is found. Learn more.
Add a Kubernetes cluster integration with a domain, or create an AUTO_DEVOPS_PLATFORM_TARGET CI variable.
Deployment strategy
Continuous deployment to production
Continuous deployment to production using timed incremental rollout
Automatic deployment to staging, manual deployment to production
Save changes

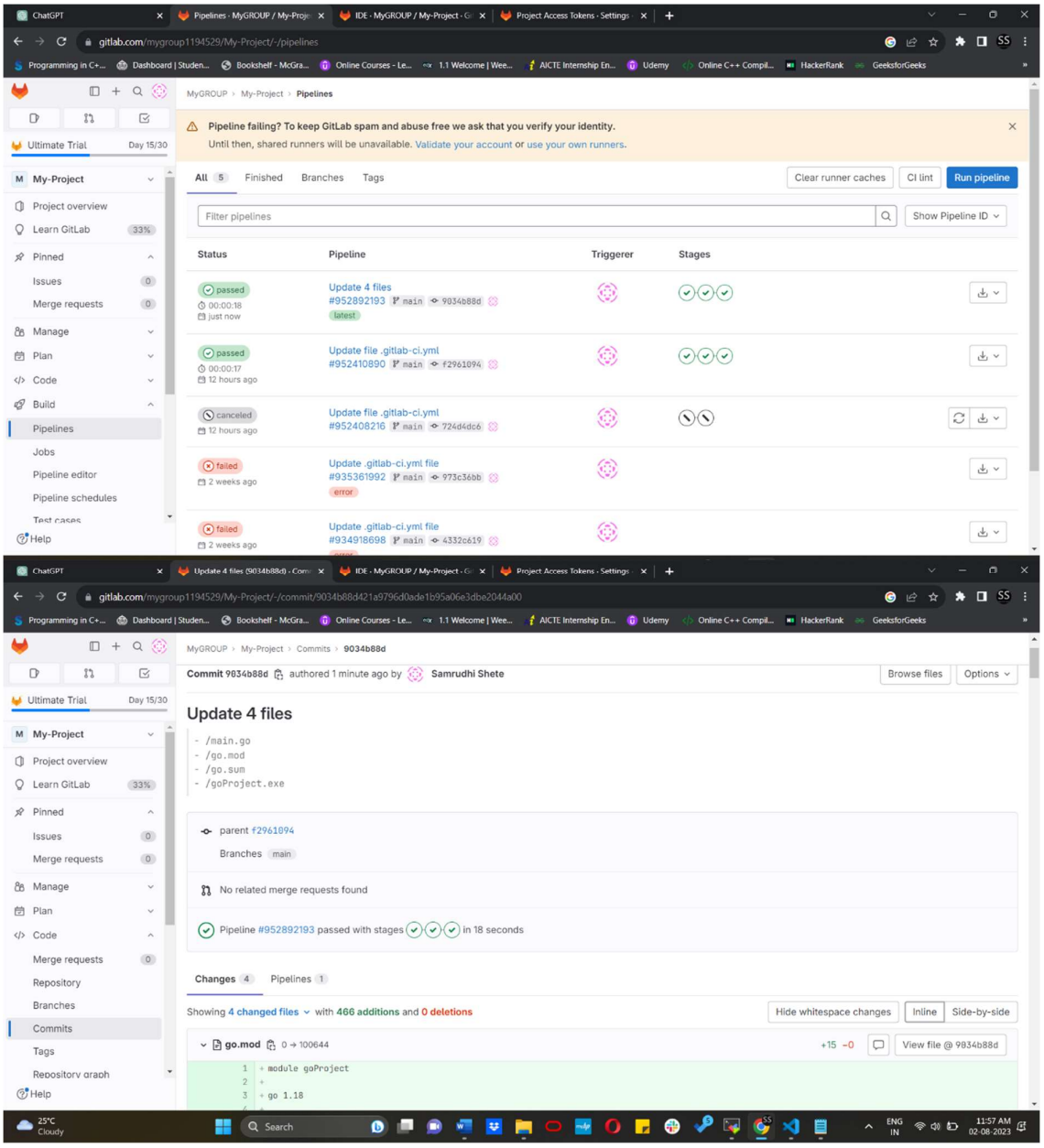

gitlab.com/mygroup1194529/My-Project/-/ci/editor
MyGROUP › My-Project › Pipeline Editor
Edit Visualize Validate NEW Full configuration
Browse CI/CD Catalog Browse templates Help
stages:
- build
- test
- deploy
build_job:
stage: build
script:
- echo "Building the project..."
No 'needs' defined, so this job doesn't have any dependencies
test_job:
stage: test
script:
- echo "Running tests..."
needs:
- build_job # This job depends on the 'build_job' to succeed before running
deploy_job:
stage: deploy
script:
- echo "Deploying the project to production..."
needs:
- test_job # This job depends on the 'test_job' to succeed before running
gitlab.com/mygroup1194529/My-Project/-/ci/editor?tab=1
Pipeline failing? To keep GitLab spam and abuse free we ask that you verify your identity.
Until then, shared runners will be unavailable. Validate your account or use your own runners.
main
Pipeline #952410890 pending for f2961094: Update file .gitlab-ci.yml
View pipeline
Pipeline syntax is correct. Learn more
Edit Visualize Validate NEW Full configuration
build
test
deploy
build_job
test_job
deploy_job

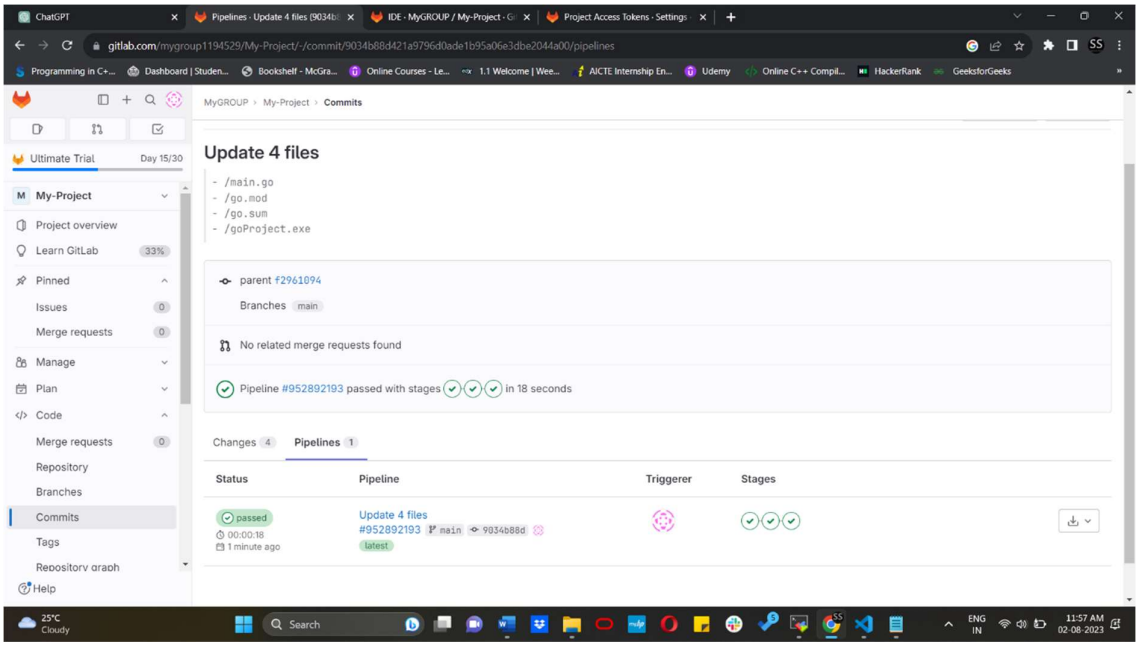

gitlab.com/mygroup1194529/My-Project/-/jobs/4784250687
MyGROUP › My-Project › Jobs › #4784250687
Pipeline failing? To keep GitLab spam and abuse free we ask that you verify your identity.
Until then, shared runners will be unavailable. Validate your account or use your own runners.
created Job test_job created 1 minute ago by Samrudhi Shete
This job has not been triggered yet
This job depends on upstream jobs that need to succeed in order for this job to be triggered
Cancel this job
test_job
Commit f2961094
Update file .gitlab-ci.yml
Pipeline #952410890 for main
test
test_job
My-Project
Project overview
Learn GitLab
Pinned
Issues
Merge requests
Manage
Plan
Code
Build
Pipelines
Jobs
Pipeline editor
Help


gitlab.com/mygroup1194529/My-Project/-/runners/25655703
MyGROUP › My-Project › CI/CD Settings › #25655703 (ftM1Avk9B)
Runner #25655703 project
Property Name
Value
Active
Yes
Protected
No
Can run untagged jobs
Yes
Locked to this project
No
Tags
ssh
Name
Version
16.2.0
IP Address
103.224.245.166
Revision
782e15da
Platform
linux
Architecture
amd64
Description
Maximum job timeout
My-Project
Project overview
Learn GitLab
Pinned
Issues
Merge requests
Manage
Plan
Code
Build
Secure
Deploy
Operate
Monitor
Help


Gitlab Runner Setup Guide
Pipeline · MyGROUP / My-Projec
Project Access Tokens · Settings
gitlab.com/mygroup1194529/My-Project/-/pipelines/952410890
MyGROUP › My-Project › Pipelines › #952410890
Update file .gitlab-ci.yml
Delete
passed Samrudhi Shete triggered pipeline for commit f2961094 finished 12 hours ago
For main
latest 3 Jobs 0 17 seconds, queued for 129 seconds
Pipeline Needs Jobs 3 Tests 0
Group jobs by Stage Job dependencies
build
test
deploy
build_job
test_job
deploy_job
Ultimate Trial Day 15/30
My-Project
Project overview
Learn GitLab 33%
Pinned
Issues 0
Merge requests 0
Manage
Plan
Code
Build
Pipelines
Jobs
Pipeline editor
Pipeline schedules
Test cases
Help
25°C Cloudy
Search
ENG IN
11:42 AM 02-08-2023


Gitlab Runner Setup Guide
Pipeline · MyGROUP / My-Projec
Project Access Tokens · Settings
gitlab.com/mygroup1194529/My-Project/-/pipelines/952410890
MyGROUP › My-Project › Pipelines › #952410890
Update file .gitlab-ci.yml
Delete
passed Samrudhi Shete triggered pipeline for commit f2961094 finished 12 hours ago
For main
latest 3 Jobs 0 17 seconds, queued for 129 seconds
Pipeline Needs Jobs 3 Tests 0
Group jobs by Stage Job dependencies
build
test
deploy
build_job
test_job
deploy_job
Ultimate Trial Day 15/30
My-Project
Project overview
Learn GitLab 33%
Pinned
Issues 0
Merge requests 0
Manage
Plan
Code
Build
Pipelines
Jobs
Pipeline editor
Pipeline schedules
Test cases
Help
25°C Cloudy
Search
ENG IN
11:42 AM 02-08-2023


gitlab.com/mygroup1194529/My-Project/-/pipelines
MyGROUP › My-Project › Pipelines
Pipeline failing? To keep GitLab spam and abuse free we ask that you verify your identity.
Until then, shared runners will be unavailable. Validate your account or use your own runners.
All 5
Finished
Branches
Tags
Clear runner caches
CI lint
Run pipeline
Filter pipelines
Show Pipeline ID
Status
Pipeline
Triggerer
Stages
passed
Update 4 files
#952892193 main 9034b88d
latest
passed
Update file .gitlab-ci.yml
#952410890 main f2961094
canceled
Update file .gitlab-ci.yml
#952408216 main 724d4dc6
failed
Update .gitlab-ci.yml file
#935361992 main 973c36bb
error
failed
Update .gitlab-ci.yml file
#934918698 main 4332c619
My-Project
Project overview
Learn GitLab
Pinned
Issues
Merge requests
Manage
Plan
Code
Build
Pipelines
Jobs
Pipeline editor
Pipeline schedules
Help
gitlab.com/mygroup1194529/My-Project/-/commit/9034b88d421a9796d0ade1b95a06e3dbe2044a00
MyGROUP › My-Project › Commits › 9034b88d
Commit 9034b88d authored 1 minute ago by Samrudhi Shete
Browse files
Options
Update 4 files
- /main.go
- /go.mod
- /go.sum
- /goProject.exe
parent f2961094
Branches main
No related merge requests found
Pipeline #952892193 passed with stages in 18 seconds
Changes 4
Pipelines 1
Showing 4 changed files with 466 additions and 0 deletions
Hide whitespace changes
Inline
Side-by-side
go.mod 0 → 100644
+15 −0
View file @ 9034b88d
1 + module goProject
2 +
3 + go 1.18
Commits
Repository
Branches
Tags



Issues: Completed Successfully

TASK 10: how to perform git local operations

Code:

- Initialize a New Repository
  git init
- Stage Files for Commit:

  # Stage all changes

  git add .

  # Stage specific files

  git add file1 file2
- Check Repository Status:

  git status
- View Commit History:

  git log
- Create a Branch:

  git branch new-branch-name
- Merge Branches:

  git merge branch-name

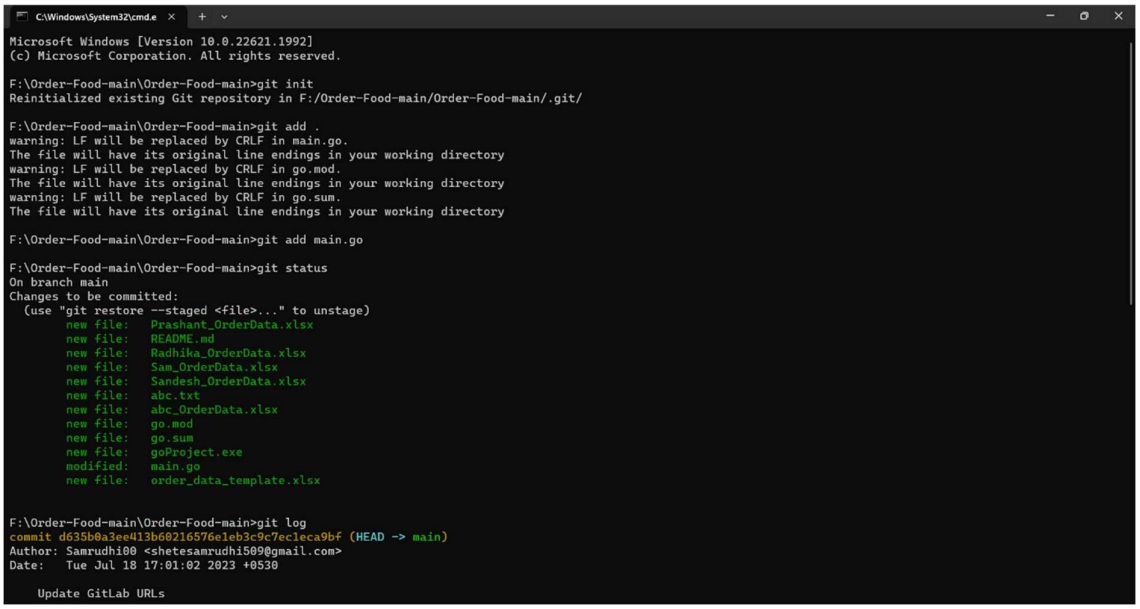
- Create and Apply Patches:

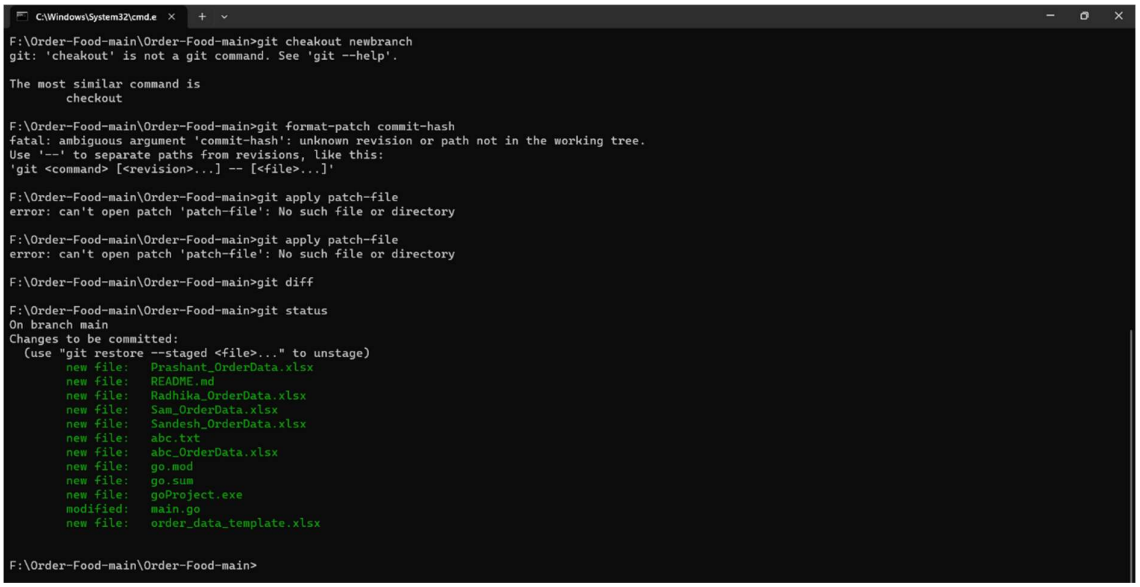
  git format-patch commit-hash # Create patch

  git apply patch-file # Apply patch
- View Differences:

  git diff # Show differences between working tree and staged changes

Output:

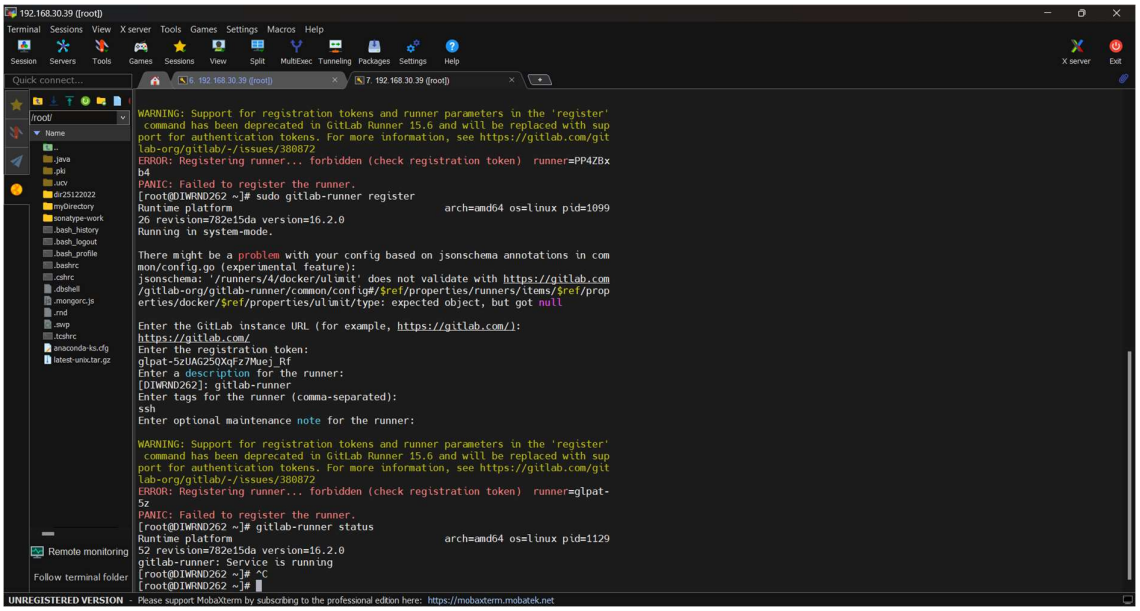
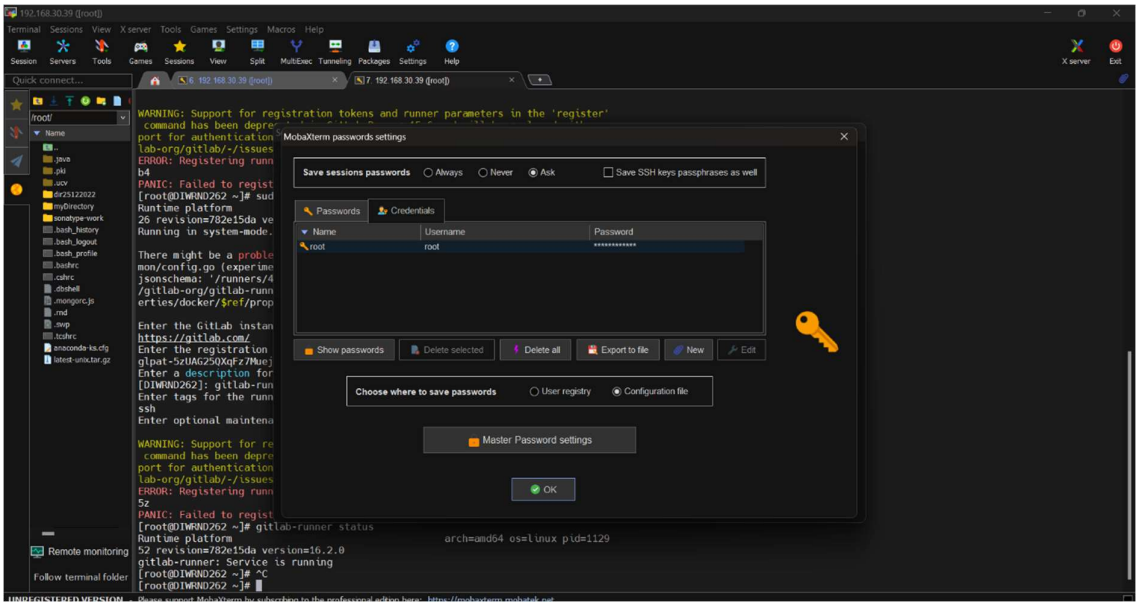




Issues: Completed Successfully

TASK 11: OpenVPN and Monotreme

1.Gitlab-runner status



2. SSH Key