

WorkFlow

1. Building the web Application using Flask , MongoDB

Requirements for running the web application are:

- Flask
- MongoDB

Pip3 install -r requirements.txt

```
agatha@agatha-VirtualBox:~/Desktop/ToDo-List-using-Flask-and-MongoDB-master$ pip3 install -r requirements.txt
Defaulting to user installation because normal site-packages is not writeable
Requirement already satisfied: flask in /home/agatha/.local/lib/python3.7/site-packages (from -r requirements.txt (line 1)) (1.0.2)
Requirement already satisfied: pymongo in /usr/local/lib/python3.7/dist-packages (from -r requirements.txt (line 2)) (3.11.0)
Requirement already satisfied: Werkzeug>=0.14 in /usr/local/lib/python3.7/dist-packages (from flask->-r requirements.txt (line 1)) (1.0.1)
Requirement already satisfied: click>=5.1 in /home/agatha/.local/lib/python3.7/site-packages (from flask->-r requirements.txt (line 1)) (7.1.2)
Requirement already satisfied: itsdangerous>=0.24 in /usr/local/lib/python3.7/dist-packages (from flask->-r requirements.txt (line 1)) (1.1.0)
Requirement already satisfied: Jinja2>=2.10 in /usr/local/lib/python3.7/dist-packages (from flask->-r requirements.txt (line 1)) (2.11.2)
Requirement already satisfied: MarkupSafe>=0.23 in /usr/lib/python3/dist-packages (from Jinja2>=2.10->flask->-r requirements.txt (line 1)) (1.1.0)
```

Start and check if mongodb is running or not

```
sudo service mongod start  
sudo systemctl status mongod
```

```
agatha@agatha-VirtualBox:~/Desktop/ToDo-List-using-Flask-and-MongoDB-master$ sudo service mongod start  
agatha@agatha-VirtualBox:~/Desktop/ToDo-List-using-Flask-and-MongoDB-master$ sudo systemctl status mongod  
● mongod.service - MongoDB Database Server  
   Loaded: loaded (/lib/systemd/system/mongod.service; disabled; vendor preset: enabled)  
   Active: active (running) since Sun 2020-10-18 13:12:47 IST; 27s ago  
     Docs: https://docs.mongodb.org/manual  
  Main PID: 5826 (mongod)  
    Memory: 199.1M  
    CGroup: /system.slice/mongod.service  
            └─5826 /usr/bin/mongod --config /etc/mongod.conf  
  
Oct 18 13:12:47 agatha-VirtualBox systemd[1]: Started MongoDB Database Server.  
agatha@agatha-VirtualBox:~/Desktop/ToDo-List-using-Flask-and-MongoDB-master$
```

Run the app.py and goto localhost



```
sudo python3 app.py
```

```
agatha@agatha-VirtualBox:~/Desktop/ToDo-List-using-Flask-and-MongoDB-master$ sudo python3 app.py
* Serving Flask app "app" (lazy loading)
* Environment: production
  WARNING: This is a development server. Do not use it in a production deployment.
  Use a production WSGI server instead.
* Debug mode: on
* Running on http://0.0.0.0:5000/ (Press CTRL+C to quit)
* Restarting with stat
* Debugger is active!
* Debugger PIN: 250-290-669
```




Goto <http://0.0.0.0:5000/> and add a task







The screenshot shows a web browser window with the address bar displaying '0.0.0.0:5000'. The page title is 'ToDo Reminder'. Below the title is a dark navigation bar with the following links: 'ALL', 'Uncompleted' (highlighted in green), 'Completed', and 'About'. The main content area displays the message 'No Tasks in the List !!' in red. Below this is a section titled 'Add a Task' in green. This section contains a form with the following fields: a text input for 'Taskname', a text area for 'Enter Description here...', a date input for 'dd / mm / yyyy', and a dropdown menu for 'Priority'. A blue 'Create' button is positioned below the form fields.

update the task

0.0.0.0:5000/update?_id=5f8bfa7f3f27cb73f338b987

ToDo Reminder

Update tasks with a reference

Unique Object ID : 5f8bfa7f3f27cb73f338b987

Task Name :

Description :

Date :

Priority :

[Return to TaskList](#)

Remove the task

0.0.0.0:5000

ToDo Reminder

ALL

Uncompleted

Completed

About

Search Reference:

Unique ID

Search Task

Search

To-Do LIST :

Status	Task Name	Description Name	Date	Priority	Remove	Modify
X	work	create the presentation	2020-10-13	Low !		

Add a Task

Taskname

Enter Description here...

dd / mm / yyyy

Priority

Create

2. Hosting the Web Application On Github Repository

Steps :

1-➤

```
git init
```

2-➤

```
git add .
```

3-➤

```
git commit -m "<Commit message>"
```

4-➤

```
git remote add origin [repository URL]
```

5-➤

```
git push -u origin branchname
```


Repository URL : <https://github.com/hkshitesh/SPCM-2020/tree/master/R171217031>

```
agatha@agatha-VirtualBox:~/Desktop/ToDo-List-using-Flask-and-MongoDB-master$ git commit -m "Simple Web Application"
[master (root-commit) 02aff4b] Simple Web Application
 22 files changed, 1253 insertions(+)
 create mode 100644 .gitignore
 create mode 100644 Dockerfile
 create mode 100644 README.md
 create mode 100644 app.py
 create mode 100644 docker-compose.yaml
 create mode 100644 k8s/deployment.yaml
 create mode 100644 k8s/service.yaml
 create mode 100644 requirements.txt
 create mode 100644 static/assets/emoji.css
 create mode 100644 static/assets/emoji.js
 create mode 100644 static/assets/style.css
 create mode 100644 static/assets/twemoji.js
 create mode 100644 static/assets/twemoji.min.js
 create mode 100644 static/images/no.png
 create mode 100644 static/images/no_xl.png
 create mode 100644 static/images/screenshot.jpg
 create mode 100644 static/images/yes.png
 create mode 100644 static/images/yes_xl.png
 create mode 100644 templates/credits.html
 create mode 100644 templates/index.html
 create mode 100644 templates/searchlist.html
 create mode 100644 templates/update.html
agatha@agatha-VirtualBox:~/Desktop/ToDo-List-using-Flask-and-MongoDB-master$
```

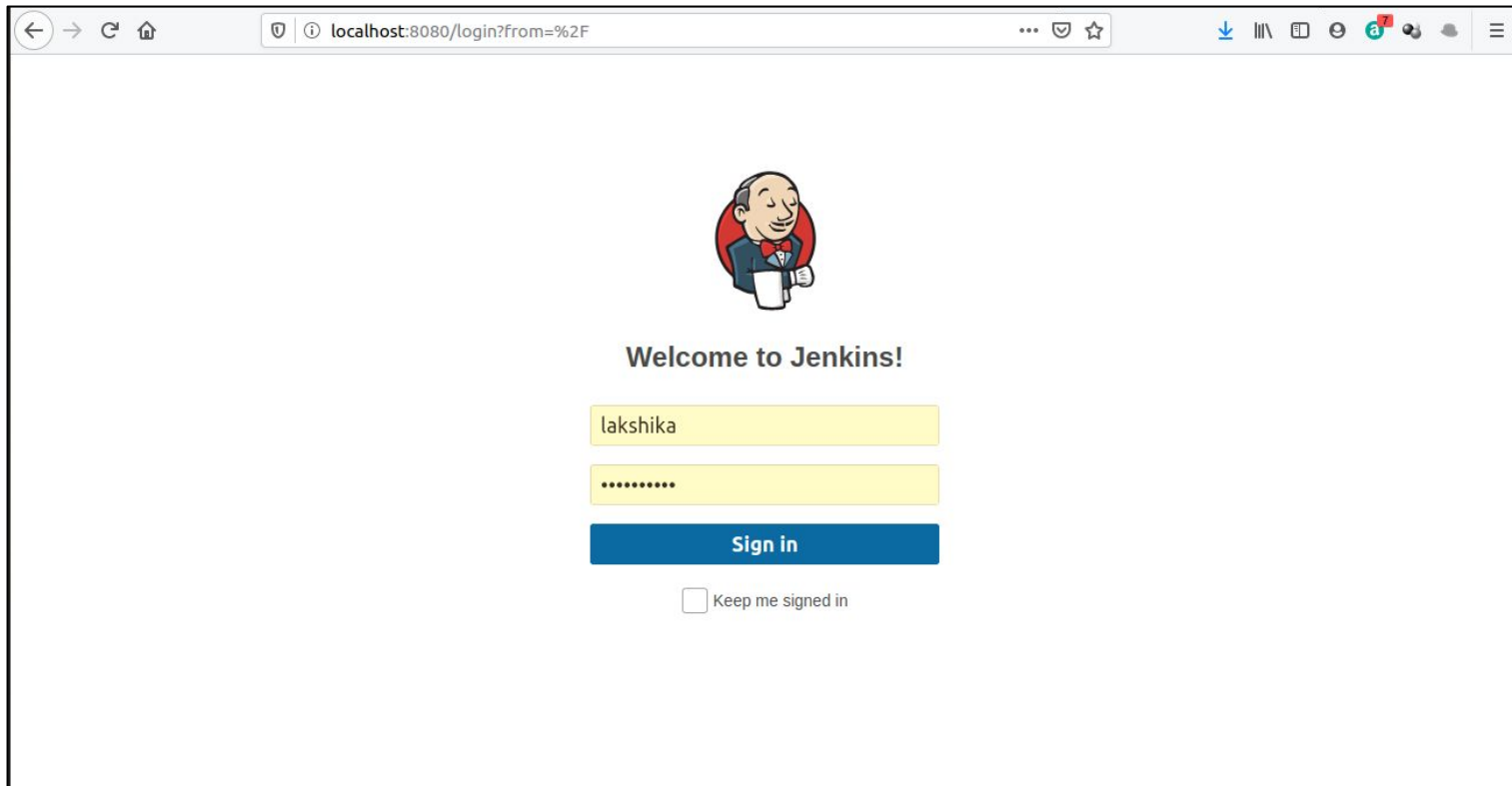
3. Setting up continuous integration build with jenkins

Check the jenkins service if it's started or not

```
agatha@agatha-VirtualBox:~/Desktop/ToDo-List-using-Flask-and-MongoDB-master$ sudo service jenkins status
[sudo] password for agatha:
● jenkins.service - LSB: Start Jenkins at boot time
   Loaded: loaded (/etc/init.d/jenkins; generated)
   Active: active (exited) since Sun 2020-10-18 12:29:47 IST; 1h 7min ago
     Docs: man:systemd-sysv-generator(8)
    Tasks: 0 (limit: 3170)
   Memory: 0B
    CGroup: /system.slice/jenkins.service

Oct 18 12:29:17 agatha-VirtualBox systemd[1]: Starting LSB: Start Jenkins at boot time...
Oct 18 12:29:36 agatha-VirtualBox jenkins[790]: Correct java version found
Oct 18 12:29:36 agatha-VirtualBox jenkins[790]: * Starting Jenkins Automation Server jenkins
Oct 18 12:29:37 agatha-VirtualBox su[1029]: (to jenkins) root on none
Oct 18 12:29:37 agatha-VirtualBox su[1029]: pam_unix(su-l:session): session opened for user jenkins by (uid=0)
Oct 18 12:29:47 agatha-VirtualBox jenkins[790]: ...done.
Oct 18 12:29:47 agatha-VirtualBox systemd[1]: Started LSB: Start Jenkins at boot time.
agatha@agatha-VirtualBox:~/Desktop/ToDo-List-using-Flask-and-MongoDB-master$
```

Goto <http://localhost:8080> in your browser



The screenshot shows a web browser window with the address bar displaying `localhost:8080/login?from=%2F`. The page content includes the Jenkins logo (a cartoon man in a tuxedo), the text "Welcome to Jenkins!", a username input field containing "lakshika", a password input field with masked characters, a blue "Sign in" button, and a checkbox labeled "Keep me signed in".

localhost:8080/login?from=%2F

Welcome to Jenkins!

lakshika

.....

Sign in

☐ Keep me signed in

Create » New Item » New Job

localhost:8080/view/all/newJob

Enter an item name

local_web_testing

» Required field

General Source Code Management Build Triggers Build Environment Build Post-build Actions

Description running the web application stored locally using jenkins

☒ Use custom workspace

Directory /home/agatha/Desktop/ToDo-List-using-Flask-and-MongoDB-master/

Display Name web-testing


☐ Keep the workspace between builds

Build

Execute shell


Command `sudo python3 app.py`

Build the job and check console output

 **Console Output**

Progress:

```
Started by user Lakshika Parihar
Running as SYSTEM
Building in workspace /home/agatha/Desktop/ToDo-List-using-Flask-and-MongoDB-master
[ToDo-List-using-Flask-and-MongoDB-master] $ /bin/sh -xe /tmp/jenkins4183916060145587717.sh
+ sudo python3 app.py
* Serving Flask app "app" (lazy loading)
* Environment: production
  WARNING: This is a development server. Do not use it in a production deployment.
  Use a production WSGI server instead.
* Debug mode: on
* Running on http://0.0.0.0:5000/ (Press CTRL+C to quit)
* Restarting with stat
* Debugger is active!
* Debugger PIN: 288-949-256
```



Now you can goto <http://0.0.0.0:5000/> and check your application

To use the sudo command in jenkins you can add this line in sudo file using the command

sudo visudo

```
# Allow members of group sudo to execute any command
%sudo  ALL=(ALL:ALL) ALL

# See sudoers(5) for more information on "#include" directives:

#includedir /etc/sudoers.d
jenkins ALL=(ALL) NOPASSWD: ALL
```

4. Containerizing the web application

```
agatha@gagatha-VirtualBox:~/Desktop/todo$ sudo docker-compose build
Building web
Step 1/4 : FROM python:2.7
2.7: Pulling from library/python
7e2b2a5af8f6: Pulling fs layer
09b6f03ffac4: Downloading [>] 81.43kB/7.812MBlling fs layer
09b6f03ffac4: Downloading [=>] 159.7kB/7.812MBiting
7e2b2a5af8f6: Pull complete
09b6f03ffac4: Pull complete
dc3f0c679f0f: Pull complete
fd4b47407fc3: Pull complete
b32f6bf7d96d: Extracting [=====>] 31.75MB/192.2MB
6f4489a7e4cf: Download complete
af4b99ad9ef0: Download complete
b32f6bf7d96d: Pull complete
6f4489a7e4cf: Pull complete
af4b99ad9ef0: Pull complete
39db0bc48c26: Pull complete
acb4a89489fc: Pull complete
Digest: sha256:cfa62318c459b1fde9e0841c619906d15ada5910d625176e24bf692cf8a2601d
Status: Downloaded newer image for python:2.7
--> 68e7be49c28c
Step 2/4 : ADD . /todo
--> b501a696a45a
Step 3/4 : WORKDIR /todo
--> Running in 1c3921250fd9
Removing intermediate container 1c3921250fd9
--> 4997d5c37aa5
Step 4/4 : RUN pip install -r requirments.txt
--> Running in 44275e2330fe
```

Run docker-compose locally and start your application


```
agatha@agatha-VirtualBox:~/Desktop/todo$ sudo docker-compose up
Starting todo_web_1 ... done
Attaching to todo_web_1
web_1 | * Serving Flask app "app" (lazy loading)
web_1 | * Environment: production
web_1 |   WARNING: This is a development server. Do not use it in a production deployment.
web_1 |   Use a production WSGI server instead.
web_1 | * Debug mode: on
web_1 | * Running on http://0.0.0.0:5000/ (Press CTRL+C to quit)
web_1 | * Restarting with stat
web_1 | * Debugger is active!
web_1 | * Debugger PIN: 262-035-408
```

```
█
```


Create a docker build to run the docker compose and push the image in docker-hub

Enter an item name

docker-push



Description

to run the docker-compose and push the image in docker hub

Git

Repositories

Repository URL

https://github.com/hkshitesh/SPCM-2020/

?

Execute shell

Command

```
cd R171217031/;  
docker build -t lakshika1064/web-application:web-app .;  
docker push lakshika1064/web-application:web-app;  
|
```

Build the job and check console output

```
Step 1/8 : FROM alpine:3.7
---> 6d1ef012b567
Step 2/8 : MAINTAINER Lakshika Parihar "pariharlakshika@gmail.com"
---> Using cache
---> 9300bf12227e
Step 3/8 : COPY . /app
---> 800f5bedb566
Step 4/8 : WORKDIR /app
---> Running in dee175008ef1
Removing intermediate container dee175008ef1
---> 0da5667ac7b8
Step 5/8 : RUN apk add --no-cache bash git nginx uwsgi uwsgi-python py2-pip      && pip2 install --upgrade pip      && pip2
install -r requirements.txt      && rm -rf /var/cache/apk/*
---> Running in ccf72339ca1b
fetch http://dl-cdn.alpinelinux.org/alpine/v3.7/main/x86\_64/APKINDEX.tar.gz
fetch http://dl-cdn.alpinelinux.org/alpine/v3.7/community/x86\_64/APKINDEX.tar.gz
(1/27) Installing pkgconf (1.3.10-r0)
(2/27) Installing ncurses-terminfo-base (6.0_p20171125-r1)
(3/27) Installing ncurses-terminfo (6.0_p20171125-r1)
(4/27) Installing ncurses-libs (6.0_p20171125-r1)
(5/27) Installing readline (7.0.003-r0)
(6/27) Installing bash (4.4.19-r1)
Executing bash-4.4.19-r1.post-install
(7/27) Installing ca-certificates (20190108-r0)
```





Tags and Scans

VULNERABILITY SCANNING - DISABLED
[Enable](#)

This repository contains 1 tag(s).

TAG	OS	PUSHED
web-app		15 hours ago

[See all](#)

Recent builds

Link a source provider and run a build to see build results here.

Readme

Repository description is empty. Click [here](#) to edit.

5. Infrastructure automation using Terraform and AWS cloud

- Aws configure
- Terraform init
- Terraform apply

```
agatha@agatha-VirtualBox:~/Desktop/ToDo-List-using-Flask-and-MongoDB-master$ sudo apt install -y awscli
[sudo] password for agatha:
Reading package lists... Done
Building dependency tree
Reading state information... Done
awscli is already the newest version (1.18.69-1ubuntu0.19.10.1).
The following packages were automatically installed and are no longer required:
 default-mysql-client gdal-data libantlr4-runtime4.7.2 libarmadillo9 libarpack2 libcharls2 libdap25 libdapclient6v5 libepsilon1 libfreexl1
 libfyba0 libgdal20 libgeos-3.7.2 libgeos-c1v5 libgeotiff2 libhdf4-0-alt libhdf5-103 libjline-java libkmlbase1 libkmldev1 libkmlengine1
 liblog4j1.2-java libminizip1 libmysqlcppconn7v5 libnetcdf13 libodbc1 libogdi4.1 libpq5 libproj13 libqhull7 libslf4j-java libspatialite7
 libsupl2 libtiff5 libxerces-c3.2 libxml2 python3-pip python3-setuptools python3-wheel python3-yaml python3-zip
Use 'sudo apt autoremove' to remove them.
agatha@agatha-VirtualBox:~/Desktop/ToDo-List-using-Flask-and-MongoDB-master$ aws configure
AWS Access Key ID [None]: AKIAJJ5B0CJH6TWKTIJA
AWS Secret Access Key [None]: XZ2QdpvdZaPGsaYA+yyy6U8Xz7Bl3CjQLhpM5iUw
Default region name [None]:
Default output format [None]:
agatha@agatha-VirtualBox:~/Desktop/ToDo-List-using-Flask-and-MongoDB-master$
```



agatha@agatha-VirtualBox: ~/Desktop/ToDo-List-using-Flask-and-MongoDB-master/tf



```
agatha@agatha-VirtualBox:~/Desktop/ToDo-List-using-Flask-and-MongoDB-master/tf$ sudo terraform init
[sudo] password for agatha:
```

Initializing the backend...

Initializing provider plugins...

- Finding latest version of hashicorp/aws...
- Finding latest version of hashicorp/template...
- Installing hashicorp/aws v3.11.0...
- Installed hashicorp/aws v3.11.0 (signed by HashiCorp)
- Installing hashicorp/template v2.2.0...
- Installed hashicorp/template v2.2.0 (signed by HashiCorp)

The following providers do not have any version constraints in configuration, so the latest version was installed.

To prevent automatic upgrades to new major versions that may contain breaking changes, we recommend adding version constraints in a `required_providers` block in your configuration, with the constraint strings suggested below.

```
* hashicorp/aws: version = "~> 3.11.0"
* hashicorp/template: version = "~> 2.2.0"
```

Terraform has been successfully initialized!

You may now begin working with Terraform. Try running `"terraform plan"` to see any changes that are required for your infrastructure. All Terraform commands should now work.

If you ever set or change modules or backend configuration for Terraform, rerun this command to reinitialize your working directory. If you forget, other commands will detect it and remind you to do so if necessary.

```
agatha@agatha-VirtualBox:~/Desktop/ToDo-List-using-Flask-and-MongoDB-master/tf$
```




agatha@agatha-VirtualBox: ~/Desktop/ToDo-List-using-Flask-and-MongoDB-master/tf



```
agatha@agatha-VirtualBox:~/Desktop/ToDo-List-using-Flask-and-MongoDB-master/tf$ sudo terraform apply
```

```
data.template_file.myapp: Refreshing state...
```

```
data.aws_availability_zones.available: Refreshing state...
```

```
data.aws_iam_policy_document.ecs_task_execution_role: Refreshing state...
```

An execution plan has been generated and is shown below.

Resource actions are indicated with the following symbols:

+ create

Terraform will perform the following actions:

```
# aws_alb.main will be created
```

```
+ resource "aws_alb" "main" {  
  + arn                        = (known after apply)  
  + arn_suffix                = (known after apply)  
  + dns_name                  = (known after apply)  
  + drop_invalid_header_fields = false  
  + enable_deletion_protection = false  
  + enable_http2              = true  
  + id                        = (known after apply)  
  + idle_timeout              = 60  
  + internal                  = (known after apply)  
  + ip_address_type           = (known after apply)  
  + load_balancer_type        = "application"  
  + name                      = "myapp-load-balancer"  
  + security_groups           = (known after apply)  
  + subnets                  = (known after apply)  
  + vpc_id                    = (known after apply)  
  + zone_id                   = (known after apply)  
}
```

```
+ subnet_mapping {  
  + allocation_id = (known after apply)  
  + outpost_id    = (known after apply)  
  + private_ipv4_address = (known after apply)  
  + subnet_id     = (known after apply)  
}
```

```
# aws_alb_listener.front_end will be created
```



agatha@agatha-VirtualBox: ~/Desktop/ToDo-List-using-Flask-and-MongoDB-master/tf



```
+ id      = (known after apply)
+ policy_arn = "arn:aws:iam::aws:policy/service-role/AmazonECSTaskExecutionRolePolicy"
+ role     = "myEcsTaskExecutionRole1064"
}
```

Plan: 9 to add, 0 to change, 0 to destroy.

Do you want to perform these actions?

Terraform will perform the actions described above.
Only 'yes' will be accepted to approve.

Enter a value: yes

aws_iam_role.ecs_task_execution_role: Creating...

aws_iam_role.ecs_task_execution_role: Creation complete after 3s [id=myEcsTaskExecutionRole1064]

aws_iam_role_policy_attachment.ecs_task_execution_role: Creating...

aws_ecs_task_definition.app: Creating...

aws_ecs_task_definition.app: Creation complete after 1s [id=myapp-task]

aws_iam_role_policy_attachment.ecs_task_execution_role: Creation complete after 3s [id=myEcsTaskExecutionRole1064-202010181535116682000000001]

aws_ecs_service.main: Creating...

aws_ecs_service.main: Creation complete after 1s [id=arn:aws:ecs:ap-south-1:262058769421:service/myapp-cluster/myapp-service]

aws_appautoscaling_target.target: Creating...

aws_appautoscaling_target.target: Creation complete after 0s [id=service/myapp-cluster/myapp-service]

aws_appautoscaling_policy.up: Creating...

aws_appautoscaling_policy.down: Creating...

aws_appautoscaling_policy.up: Creation complete after 1s [id=myapp_scale_up]

aws_appautoscaling_policy.down: Creation complete after 1s [id=myapp_scale_down]

aws_cloudwatch_metric_alarm.service_cpu_high: Creating...

aws_cloudwatch_metric_alarm.service_cpu_low: Creating...

aws_cloudwatch_metric_alarm.service_cpu_low: Creation complete after 1s [id=myapp_cpu_utilization_low]

aws_cloudwatch_metric_alarm.service_cpu_high: Creation complete after 1s [id=myapp_cpu_utilization_high]

Apply complete! Resources: 9 added, 0 changed, 0 destroyed.

Outputs:

alb_hostname = myapp-load-balancer-1465829641.ap-south-1.elb.amazonaws.com

agatha@agatha-VirtualBox: ~/Desktop/ToDo-List-using-Flask-and-MongoDB-master/tf\$