6/3/25

Program 2

Install jdk 17/21
copy bin path to environment variables
Check for java version in cmd (java --version
java -version)

Install jenkins
Q Download Jenkins
In link check for software requirement to comply with the jdk
Select left one -> click on windows
test 8080 ✓ -> change path of jdk (if done, leave)
☒ change to local hard disk (first option)

github.com/techworld with nana/web-applicati

Then type localhost:8080
Set the things

Install Tomcat
Q tomcat 9 download
~~Go~~ Install it and give port 8085

In localhost:8080 -> Jenkins
Dashboard -> + New item -> Name "Deploy a war file"
click 'Free style Project'
OK ✓

Now config tab opens -> click Git
Paste the github repository link (if private give credentials)

If error

Dashboard -> Manage Jenkins -> Plugins
Available plugins -> GitHub
Maven Integration
Deploy to container
} install

-> Manage Jenkins -> Tools (Go down)
-> For maven installation
name:-> maven <version>

Now in views -> Select yours -> Configure
paste the git link
Build steps -> invoke top level maven ->
Maven Version - maven 3.9.9
Goals - clean install

Post build actions -> Deploy war/ear to container ->
WAR/EAR files : **/*.war
containers -> Tomcat 9
Tomcat URL: http://localhost:8085

Give tomcat username & password
(add credentials)
Jenkins username & password

In C://Programfiles/apache.../tomcat 9.0/conf/tomcatusers type (replace)

```
<role rolename="manager-gui"/>
<role rolename="admin-gui"/>
<user username="tomcat" password="tomcat"
      roles="manager-gui", admin-gui, manager-script,
             manager-jmx, manager-status"/>
```

13/3/25

If localhost doesnt load
Q services in Windows
Select tomcat and start

(In eclipse -> New -> File -> Maven project ->)

If deploy war/ear to container - BUILD FAILURE
Go to installed plugins -> ⊙ (disable) should be grey

13/3.

In eclipse -> New -> File -> Maven project -> ☑ Simple project -> Next
Group id: com.dsce
artifact id: pgrm2 (project name)
packaging: war
-> Finish

Now the program structure opens

In pgrm2 -> In src folder -> In main folder -> Inside webapp folder
-> Create a new folder
Name: WEB-INF
-> Finish

-> Create a new file
Name: index.html
-> Finish

In Index.html type

```
<html>
  <body>
    <h1>Hello World</h1>
  </body>
</html>
```

Now in WEB-INF folder -> Create a new file
Name: web.xml
-> Finish

In web.xml type

```
<plugin>
<groupID>org.apache.maven.plugins</groupId>
<artifactId>maven-war-plugin</artifactId>
<configuration>
          normal
<webXml>src/main/webapp/WEB-INF/index.html</webXml)
</configuration>
</plugin>
```

Here is path, copy path of WEB-INF from properties and paste the name of the file created
The html filename must match with the path in above code

Right click on project (pgrmd) -> Maven -> Update Project
-> OK

-> Run As -> Maven Clean
-> Maven install

After BUILD SUCCESS
Check in pgrmd/target folder -> pgrmd-0.0.1-SNAPSHOT.war
has been generated
i.e., the war file has been generated

Now Right click pgrmd -> Run as -> Maven build
goals: compile
-> Run
(BUILD SUCCESS)

~~Now in Github~~
Now in cmd of system:-
> cd <path of project> (check properties of pgrmd & copy whole path
> git init

Now again in eclipse:-
Right click on pgrmd -> Team -> Share Project ->
(Now the git is initialized) Finish
If git is not initiated, it asks to create new repository

In github
-> Create new repository -> Give a Name
(DevopsWar)
-> Keep the https link (URL)
(If needed a token for password, settings -> Developer settings
personal access tokens -> Tokens (classic)
-> Generate new Token (classic)
Give a name and select all the scopes ☑
-> Generate Token
This can be used as password later)

In eclipse
Right click pgrmd -> Team -> Add to index
-> Commit
Type some msg & click commit
(If error to repository comes perform next step)

-> Team -> Remote -> Push -> Paste the repository URL

->

src ref: master
Click All Branches -> select force update
-> Finish

Enter Github username & password (might ask twice)
-> Finish -> Close
Enter commit msg -> commit and push

If error -> connect to repository
Click push head -> paste Git URL -> Preview
-> Enter username and password (might ask twice)

Finish -> Close

Now in Github, Refresh the page, the files should be loaded. Copy this repository link and paste in Jenkin's Git field and do the same as before

If successful build
Goto to Tomcat -> Manager Server -> Enter username & password
Click on the war file name generated
The index.html opens -> Success