DevOps Lab viva questions ananswers

Experiment 1: Write code for a simple user registration form for an event

Viva Questions & Answers

1. Q: What is the purpose of this experiment?

A: To design and implement a simple user registration form using Flask and HTML.

2. Q: Which framework is used in the backend?

A: Flask (a lightweight Python web framework).

3. Q: Why do we use Flask here?

A: Flask is simple, minimal, and good for small applications and prototypes.

4. Q: What is the role of app.py?

A: It is the main backend script that handles routes, requests, and responses

5. Q: What does the render_template() function do?

A: It loads and renders an HTML template from the templates/ folder.

6. Q: What is the use of the request object?

A: It captures user inputs from forms (like username, email, password).

7. Q: Why do we use redirect(url_for('success'))?

A: To navigate to the success page after form submission.

8. Q: What is __name__ == '__main__' in Flask?

A: It ensures that the app runs only when executed directly, not when imported.

9. Q: What is the role of the users list in the code?

A: It temporarily stores registered user details (demo purpose only).

10. Q: Why don't we store the password in users?

A: For security reasons – passwords should be encrypted and stored in a database.

11. Q: What does debug=True do in Flask?

A: It enables auto-reloading and debugging error messages.

12. Q: Which file contains dependencies of the project?

A: requirements.txt.

13. Q: What is the purpose of Docker in this experiment?

A: To containerize the Flask app so it can run consistently across environments.

14. Q: Which base image is used in the Dockerfile?

A: python:3.9-slim.

15. Q: What is the command to build the Docker image?

A: docker build -t exp1 .

16. Q: What is the command to run the container?

A: docker run -p 5000:5000 exp1

17. Q: Why do we expose port 5000 in Dockerfile?

A: Because Flask runs on port 5000 by default.

18. Q: How do we access the registration form after running the container?

A: By visiting http://localhost:5000/register in a browser.

19. Q: What is the advantage of using Docker here?

A: It ensures portability, consistency, and easy deployment of the app.

20. Q: How would you extend this experiment in a real project?

A: Replace temporary storage with a database (e.g., MySQL, PostgreSQL) and implement user authentication.

Experiment 2: Explore Git and GitHub Commands

Viva Questions & Answers

1. Q: What is Git?

A: Git is a distributed version control system used to track changes in code.

2. Q: What is GitHub?

A: GitHub is a cloud platform for hosting Git repositories and collaborating on projects.

3. Q: Difference between Git and GitHub?

A: Git is a version control tool, GitHub is a hosting platform for Git repositories.

4. Q: Command to initialize a Git repository?

A: git init

5. Q: Command to check the current status of repository?

A: git status

6. Q: Command to add files to staging area?

A: git add <filename> or git add .

7. Q: Command to commit changes?

A: git commit -m "message"

8. Q: Command to view commit history?

A: git log

9. Q: Command to undo changes in a file?

A: git checkout <filename>

10. Q: Command to create a new branch?

A: git branch <branch-name>

11. Q: Command to switch between branches?

A: git checkout <branch-name>

12. Q: Command to merge two branches?

A: git merge <branch-name>

13. Q: Command to push changes to GitHub?

A: git push origin <branch-name>

14. Q: Command to clone a GitHub repo?

A: git clone <repo-url>

15. Q: What is a Pull Request in GitHub?

A: A request to merge changes from one branch into another on GitHub.

16. Q: What is the difference between git pull and git fetch?

A: git pull updates local repo immediately, git fetch only downloads changes.

17. Q: What is a remote in Git?

A: A reference to a remote repository, usually hosted on GitHub/GitLab.

18. Q: Command to add a remote repo?

A: git remote add origin <url>

19. Q: What is the default branch in GitHub?

A: Usually main (earlier master).

20. Q: Why is Git important in DevOps?

A: It enables collaboration, versioning, CI/CD integration, and rollback options.

Experiment 3: Practice Source Code Management on GitHub (using Exp-1 code):: Viva Questions & Answers

1. Q: What is source code management (SCM)?

A: It is the practice of tracking and controlling changes to source code using tools like Git.

2. Q: Why do we use GitHub for SCM?

A: It provides collaboration, remote backup, versioning, and CI/CD integration.

3. Q: Command to clone a GitHub repo?

A: git clone <repo-url>

4. Q: Command to check which branch you are in?

A: git branch

5. Q: Command to move into a cloned repository folder?

A: cd <repo-name>

6. Q: Command to stage all files in the repo?

A: git add .

7. Q: Command to commit files with a message?

A: git commit -m "message"

8. Q: Command to push changes to GitHub?

A: git push origin main

9. Q: What is a README file?

A: A documentation file explaining project details, usually in Markdown.

10. Q: What is the .gitignore file used for?

A: To ignore files/folders that shouldn't be tracked by Git.

11. Q: What is the difference between local repo and remote repo?

A: Local repo is on your computer, remote repo is hosted on GitHub.

12. Q: Command to connect local repo to remote repo?

A: git remote add origin <url>

13. Q: Why is branching important?

A: It allows parallel development without disturbing the main branch.

14. Q: What is a fork in GitHub?

A: A personal copy of another user's repository.

15. Q: What is a pull request (PR)?

A: A request to merge code from one branch/repo into another.

16. Q: What is GitHub Actions?

A: A CI/CD tool provided by GitHub for automation of builds and tests.

17. Q: Command to check the last commit details?

A: git show

18. Q: What is the difference between git fetch and git pull?

A: fetch downloads changes, pull downloads and merges them.

19. Q: Why should developers regularly push code to GitHub?

A: To back up work, collaborate, and avoid conflicts.

20. Q: How does GitHub support DevOps?

A: It integrates with CI/CD pipelines, issue tracking, and automation workflows.

Experiment 4: Jenkins Installation and Setup

Viva Questions & Answers

1. Q: What is Jenkins?

A: An open-source automation server used for Continuous Integration/Delivery (CI/CD).

2. Q: What is the default port of Jenkins?

A: Port 8080.

3. Q: What language is Jenkins written in?

A: Java.

4. Q: How do you access Jenkins after installation?

A: Open http://localhost:8080 in a browser.

5. Q: What is the first step after installing Jenkins?

A: Unlock Jenkins using the initial admin password.

6. Q: What are Jenkins plugins?

A: Extensions that add extra features to Jenkins (e.g., Git, Docker, Kubernetes plugins).

7. Q: What is a Jenkins job?

A: A task or project configured in Jenkins to build, test, or deploy applications.

8. Q: What is a Jenkins pipeline?

A: A script or workflow that defines automated steps for building and deploying applications.

9. Q: Where is Jenkins configuration stored?

A: In XML files inside the Jenkins home directory.

10. Q: What is Jenkins Master?

A: The main Jenkins server that manages jobs and coordinates build agents.

11. Q: What is a Jenkins Agent (Node)?

A: A separate machine that executes build tasks assigned by the master.

12. Q: What is the purpose of installing recommended plugins during setup?

A: To quickly enable common CI/CD features without manual setup.

13. Q: Can Jenkins integrate with GitHub?

A: Yes, using the GitHub plugin for source code management.

14. Q: How to restart Jenkins safely?

A: Run: http://localhost:8080/safeRestart

15. Q: What is Jenkins CLI?

A: A command-line interface to manage Jenkins remotely.

16. Q: How do you secure Jenkins?

A: By enabling authentication, authorization, and role-based access.

17. Q: What is Jenkinsfile?

A: A text file that defines a pipeline as code.

18. Q: What operating systems support Jenkins?

A: Windows, Linux, macOS (cross-platform).

19. Q: What is Continuous Integration (CI)?

A: The practice of automatically building/testing code after every commit.

20. Q: Why is Jenkins important in DevOps?

A: It automates build, test, and deployment, improving software delivery speed.