GitHub Actions – Complete Beginner Guide (PDF)

1. What is GitHub Actions?

GitHub Actions is an automation tool built into GitHub that allows developers to automatically run tasks when certain events happen in a repository. These tasks can include:

- Building code
- Testing code
- Deploying applications
- Running scripts
- Automating workflows

GitHub Actions works using simple YAML files stored inside your project.

---

2. Why is GitHub Actions Used?

GitHub Actions is used because it automates repetitive tasks. It supports:

✔ Continuous Integration (CI)

Automatically tests and builds your application every time code is pushed.

✔ Continuous Deployment (CD)

Automatically deploys your project to a server or hosting platform.

✔ Scheduled Automation

Run tasks daily/weekly/monthly using cron jobs.

✔ Manual Triggers

Run workflows with a button click using workflow_dispatch.

---

3. Is GitHub Actions CI/CD?

Yes.

GitHub Actions can be used for:

- **CI (Continuous Integration)** → Build + test code automatically
- **CD (Continuous Deployment)** → Deploy to server automatically

GitHub Actions = Full CI/CD platform.

---

4. How GitHub Actions Works

1. You push code to GitHub.
2. GitHub detects the push.
3. GitHub Actions reads your YAML workflow.
4. GitHub creates a virtual machine (like Ubuntu).

5. It runs the steps mentioned in your YAML file.
6. Shows results in the Actions tab.

---

5. Folder Structure Required
Your YAML workflow file must be placed here:

```
your-project/
.github/
workflows/
my-workflow.yml
```

GitHub only reads workflows from this folder.

---

6. Basic Workflow Example
A simple workflow that prints text:

```yaml
name: Echo Workflow

on:
push:
branches: [ main ]
workflow_dispatch:

jobs:
echo_job:
runs-on: ubuntu-latest

steps:
- name: Checkout code
uses: actions/checkout@v4

- name: Echo sample text
run: echo "Hello from GitHub Actions!"
```

---

7. Steps to Set Up GitHub Actions in Your Project

Step 1 — Create a project folder

```
my-project/
```

Step 2 — Create the workflow folder
```
my-project/.github/workflows/
```

Step 3 — Add a YAML file
```
my-project/.github/workflows/deploy.yml
```

Step 4 — Initialize Git
```
git init
git branch -M main
git add .
git commit -m "Initial commit"
```

Step 5 — Add GitHub remote
```
git remote add origin
```

Step 6 — Push to GitHub
```
git push -u origin main
```

Step 7 — Go to GitHub → Actions
Your workflow will run automatically.

---

8. Real-World Use Cases
✔ Deploy a React app
✔ Deploy a Django backend
✔ Run tests automatically
✔ Build Docker images
✔ Run Python scripts daily

✔ Automated backups

✔ Code quality checks

---

9. Example: Deployment Workflow

```yaml
name: Deploy App

on:
push:
branches: [ main ]

jobs:
deploy:
runs-on: ubuntu-latest
steps:
- uses: actions/checkout@v4

- name: Install dependencies
run: npm install

- name: Build project
run: npm run build

- name: Deploy to server
run: echo "Deploying..."
```

---

10. Why Developers Use GitHub Actions

- Reduces manual work
- Avoids mistakes in deployment
- Standardizes development workflow
- Saves time in development
- Essential skill in modern DevOps
- Connects directly with GitHub repositories
- Free for most users

---

11. Summary

GitHub Actions is one of the most powerful automation tools available today. It allows you to automate builds, tests, deployments, and scheduled tasks using simple YAML files. Understanding GitHub Actions is essential for modern development, DevOps, and cloud deployment workflows.

This PDF covered:

- What is GitHub Actions
- Why it is used
- Whether it is CI/CD
- How to set up a project
- Basic workflow structure
- Real-world examples