**SCRIPT MODULE 7**

**Topic 1: What is GitHub Actions?**

"GitHub Actions is like your personal robot assistant inside GitHub. It lets you automate things like testing code, formatting Markdown, deploying your app, and more — every time someone pushes or pulls code. It's built around the idea of *workflows* that respond to *events*."

✅ **No hands-on here yet. This is conceptual.**

**Topic 2: CI/CD Concepts (High-Level)**

Imagine a team of students working on a group project. Every student is writing parts of a report on their own computer. To avoid problems, they decide:

* Everyone must **upload their part** to a shared Google Doc **only after checking for errors** (CI).
* Once everything is reviewed and approved, the final report is automatically **sent to the teacher** without manual effort (CD).

**Continuous Integration (CI)**

* **Goal**: Make sure every change added to the code is clean, bug-free, and doesn’t break anything.
* **How it works**: When a developer pushes code or creates a Pull Request, GitHub Actions automatically runs checks:
  + Runs tests
  + Lints code
  + Checks for errors
* **Think of it like**: Grammarly automatically checking your essay every time you write a new paragraph.

"CI ensures everyone on the team is contributing error-free code. Every time you make changes and push them, GitHub Actions automatically checks the quality of your work — like a teacher marking your homework instantly!"

**Continuous Delivery / Deployment (CD)**

* **Goal**: Automatically **deliver** the latest working version of your app to users (or to a testing area).
* **Delivery** = Code is ready to be deployed manually.
* **Deployment** = Code is automatically pushed live (e.g., to a website or app store).
* **Think of it like**: Once your group report passes all checks, it’s automatically submitted to the teacher without anyone clicking "Send."

"CD automates the final step — taking your working project and shipping it to production. With GitHub Actions, we can automate deployment so our website updates automatically when we push new code."

**GitHub Actions Role in CI/CD**

* You define workflows using .yml files in .github/workflows/
* These workflows run automatically on triggers like:
  + Pushes
  + Pull Requests
  + Releases
* Common tasks:
  + Run tests
  + Deploy to servers
  + Notify teams
  + Lint Markdown or Python files

CI/CD automates code quality checks and deployments.

CI = auto-check your code (like Grammarly for developers).

CD = auto-deploy your app (like auto-submitting an assignment).

GitHub Actions makes CI/CD easy with ready templates.

Widely used in jobs for faster, safer software delivery.

"CI/CD stands for **Continuous Integration** and **Continuous Deployment/Delivery**. Think of it as this:

* CI means 'test everything as soon as someone pushes code'.
* CD means 'deploy it automatically once it passes tests'.

GitHub Actions is a free way to implement this in small and large projects without any third-party tools."

✅ **Conceptual only — no command line yet.**

**Topic 3: Use Workflows to Auto-Check PRs or Run Tests**

Step 1: Clone your GitHub repo

git clone https://github.com/YourUsername/automation-demo.git

cd automation-demo

Step 2: Create a README.txt file

echo "This is a test file. Say hello to GitHub Actions!" > README.txt

✅ Make sure it contains the word “hello”.

Step 3: Create a GitHub Actions workflow

Create the folder:

mkdir -p .github/workflows

Create the file:

touch .github/workflows/check-readme.yml

Step 4: Commit & Push the changes

git add .

git commit -m "Add README checker workflow"

git push origin main

**Step 5: View the results on GitHub**

1. Go to your GitHub repo online.
2. Click the **"Actions"** tab.
3. Click on the latest workflow run.
4. You’ll see either:
   * ✅ Successful run with “README.txt contains 'hello'”
   * ❌ Failed run if the keyword was missing

To Test Failure

Change the file:

echo "Just some random text." > README.txt

git add README.txt

git commit -m "Test: Remove 'hello'"

git push

"Let’s imagine we want to make sure that every repo has a proper README, and it mentions something important — like the word 'hello'.

With GitHub Actions, we can write a small workflow that checks this for us. If the file doesn’t meet our requirement, GitHub will stop the pull request or push — just like a robot quality checker.

This is the start of understanding CI/CD — using automation to reduce human error and keep code consistent."

"When you raise a Pull Request, this action automatically runs and highlights formatting issues. This is called *pre-merge validation* — a key feature of modern CI pipelines."

**Topic 4: Templates to Get Started**

**Speaker Notes:**

"GitHub provides many pre-built templates you can just drop into your repo. For example:

* Python test runners
* Node.js CI
* Docker builds
* Deploy to Firebase, Netlify, Heroku

You can also use the GitHub Marketplace to explore community workflows."

**➤ Step-by-Step: Use a Template from GitHub Actions**

1. Go to your repo → Click **Actions tab**
2. Choose a suggested workflow:
   * e.g., **Node.js CI**, **Python Lint**, or **Markdown Lint**
3. GitHub auto-fills a workflow for you
4. Click **“Start commit” → Commit new file**

"This is the fastest way to get started. GitHub reads your project and recommends appropriate workflows. These templates save time and make automation easy, even for beginners."

**Summary Table**

| **Feature** | **Description** |
| --- | --- |
| **Workflow** | A YAML file that defines what should happen (the automation) |
| **Trigger** | Event like push, pull\_request, issue, or even scheduled |
| **Job** | A set of steps to run (like a script) |
| **Runner** | A virtual machine (Ubuntu, Windows) where your code runs |
| **Marketplace** | Public library of pre-built GitHub Actions |

**Real-Life Analogy**

| **GitHub Action Feature** | **Analogy** |
| --- | --- |
| on: push trigger | Like setting your house lights to turn on every time you open the door |
| Workflows | Your home automation rules |
| Marketplace Actions | Smart appliances you plug into your rules |
| CI/CD | Quality check and delivery guy working together, automatically |