# **Workflow Documentation**

This document provides detailed explanations for each GitHub Actions workflow used in this project, including their configuration, purpose, and how to interpret their results.

## 1 Continuous Integration (ci.yml)

**Purpose:**  
Ensures that code changes are linted, tested, and validated before merging into the main branch.

**Configuration:**  
- Trigger: On every push or pull request to main. - Steps: - Checkout repository - Setup Node.js environment - Install dependencies - Lint and test the code

**Interpreting Results:**  
A tick means the build passed successfully.  
A cross indicates failed linting or tests — open the workflow run logs to see which step failed.

**Artifacts / Reports:**  
📄 *Placeholder for CI test reports and logs.*

## 2 Deployment Workflow (deploy.yml)

**Purpose:**  
Automatically deploys the app (for example, to Vercel) after CI passes.

**Configuration:**  
- Trigger: On push to main. - Steps: - Build project - Deploy using Vercel CLI

**Interpreting Results:**  
- A tick mean Successful deployment  
- A cross means Deployment error — usually caused by misconfigured tokens or missing environment variables.

**Artifacts / Reports:**  
📄 *Placeholder for deployment logs and screenshots.*

## 3 Scheduled Tasks (scheduled.yml)

**Purpose:**  
Runs background maintenance tasks (e.g., cleaning old data, updating satellite data) on a schedule.

**Configuration:**  
- Trigger: Daily (cron job) - Can run any script like npm run maintenance

**Interpreting Results:**  
Check workflow logs under the “Scheduled Tasks” name in the Actions tab.

**Artifacts / Reports:**  
📄 *Placeholder for maintenance logs.*

## 4 Dependency Updates (dependency-updates.yml)

**Purpose:**  
Automatically checks for new versions of dependencies and opens pull requests to update them.

**Configuration:**  
- Uses Dependabot. - Triggers weekly.

**Interpreting Results:**  
- Pull requests created for outdated dependencies.  
- Conflicts indicate version incompatibilities.

**Artifacts / Reports:**  
📄 *Placeholder for dependency update logs.*

## 5 Code Review Workflow (code-review.yml)

**Purpose:**  
Automates static analysis, linting, and security scans on pull requests.

**Configuration:**  
- Trigger: On pull request creation or updates. - Runs ESLint, CodeQL, and security scanning tools.

**Interpreting Results:**  
- A tick indicates Passed: code meets style and security standards.  
- A cross indicates Failed: check the “Annotations” tab in the pull request for detailed issues.

However, it is important to note that since the repository has been archived by the owner, pull requests are automatically blocked so I wasn’t able to test this.

**Artifacts / Reports:**  
📄 *Placeholder for code review summaries and scan results.*

## 6 Documentation Deployment (docs-deploy.yml)

**Purpose:**  
Builds and deploys project documentation automatically to GitHub Pages.

**Configuration:**  
- Trigger: On changes to /docs or manual dispatch. - Uses docsify or mkdocs to generate documentation.

**Interpreting Results:**  
- A tick indicates Documentation successfully deployed to GitHub Pages.  
- A cross indicates Build or deploy error — check workflow logs for missing dependencies or bad markdown.

**Artifacts / Reports:**  
📄 *Placeholder for deployment logs and screenshots.*

## 🗾 Summary

| Workflow | Trigger | Output |
| --- | --- | --- |
| CI | Push / PR | Test + Lint results |
| Deployment | Push to main | Deployed app |
| Scheduled Tasks | Cron | Maintenance logs |
| Dependency Updates | Daily | Updated dependencies |
| Code Review | PR | Lint / Security results |
| Docs Deploy | Docs change | Live documentation |