**Setup of Development, Testing, and Staging Environments**

**Objective:**  
Establish isolated, scalable, and automated environments for safe coding, quality assurance, and pre-production validations — ensuring continuous delivery and high-quality releases.

**Detailed Components:**

* **Environment Configuration:**
  + Separate **Dev**, **QA/Test**, and **Staging** environments mirroring production setup.
  + Isolated databases and config variables for each tier.
* **CI/CD Pipeline Implementation:**
  + Tools: GitHub Actions / GitLab CI / Jenkins / Azure DevOps.
  + Automated workflows for:
    - Code integration and unit testing (on every push).
    - Build artifacts creation (e.g., APKs/IPA).
    - Deployment to test or staging environments.
* **Containerization & Orchestration:**
  + Docker-based containers for services (API, auth, DB, etc.).
  + Kubernetes (K8s) for orchestration, if using large-scale architecture.
* **Test Automation Integration:**
  + Integration with tools like Selenium, Appium, or Postman.
  + Environment-specific test data sets and seeding scripts.
* **Monitoring & Debugging Tools:**
  + Application logging (e.g., LogRocket, ELK stack).
  + Error tracking and crash reporting tools (e.g., Sentry, Firebase Crashlytics).