Documentation

Phase 2: Project Planning and GitHub Setup

1. GitHub Issues

Created structured issues to break down work:

- Create login screen
- Implement Firebase authentication
- Design daily water consumption UI
- Integrate Firebase Realtime Database

Purpose: Tracks task and improves sprint planning.

2. GitHub Project Board

- Created SDG6 Mobile App DevOps Board
- Columns: To Do, In Progress, Done
- Linked issues for visual tracking

Purpose: Visualize development progress with issue tracking.

3. Git Branching Strategy

- ❖ Created a dev branch for development
- ❖ Created feature branches: feature/login-screen, etc.

```
bash
CopyEdit
git checkout -b dev
git pushes origin dev

git checkout -b feature/login-screen
```

Purpose: Supports collaboration and clean development.

4. Pull Request Template

`.github/PULL_REQUEST_TEMPLATE.md`:

```
markdown
CopyEdit
## 📝 Summary
Briefly explain what your PR does.

## ✅ Checklist
* [ ] Feature implemented
* [ ] Code builds without errors
* [ ] Tests added or updated
* [ ] Related documentation updated
```

Purpose: Enforces clean pull request practices.

5. Contributing Guide

`CONTRIBUTING.md`:

```
markdown
CopyEdit
# 🚀 Contributing Guide

## Git Workflow
- Use the `dev` branch for development.
- Create a branch: `feature/your-feature-name`.
- Submit pull requests to merge into `dev`.

## Commit Format (Conventional Commits)
- feat: New feature
- fix: Bug fix
- docs: Documentation
```

Phase 3: CI/CD and Quality Assurance

6. GitHub Actions for Android Build

`.github/workflows/android.yml`:

```yaml
yaml
CopyEdit
name: Android CI
on:
  push:
    branches: [dev]
  pull_request:
    branches: [dev]

jobs:
  build:
    runs-on: ubuntu-latest
    steps:
      - uses actions/checkout@v3
      - uses actions/setup-java@v3
        with:
          java-version: '11'
      - run:/gradlew build
```

Purpose: Automates app build validation.

7. Basic Instrumented Test

File: `src/androidTest/java/.../ExampleInstrumentedTest.java`

```java
java
CopyEdit
@RunWith (AndroidJUnit4.class)
public class ExampleInstrumentedTest {
```

```
  @Test
  public void useAppContext() {
    Context appContext =
InstrumentationRegistry.getInstrumentation().getTargetContext();
    assertEquals("com.your.package.name",
appContext.getPackageName());
  }
}
```

8. Security with Code Scanning

- **Enable Dependabot** in GitHub:
- Settings → Security → Enable Dependabot alerts

Add CodeQL Workflow: `.github/workflows/codeql-analysis.yml`

```
yaml
CopyEdit
name: "CodeQL"

on:
  push:
    branches: [dev]
  pull_request:
    branches: [dev]
  schedule:
    - cron: '0 0 * * 0'

jobs:
```

```
  analyze:
    runs-on: ubuntu-latest
    strategy:
      matrix:
        language: ['java']
    steps:
      - uses: actions/checkout@v3
      - uses: github/codeql-action/init@v2
        with:
          languages: ${{ matrix.language }}
      - uses: github/codeql-action/autobuild@v2
      - uses: github/codeql-action/analyze@v2
```

Phase 4: Production Deployment & Observability

9. Release Strategy

```
bash
CopyEdit
git tag -a v1.0.0 -m "Initial Release"
git pushes origin v1.0.0
```

- Create a release on GitHub → Link the tag.

10. Observability

Firebase Crashlytics Integration (optional)

Google Analytics for app usage events

Logcat during testing to capture runtime logs

Phase 5: Documentation and Operational Excellence

11. Divio-Style Documentation

Create a /docs folder with:

- tutorials/ → How to install the app
- how-to-guides/ → "How to track water usage"
- reference/ → Firebase DB structure, SDKs used
- explanation/ → Why Firebase? Why CI/CD?

12. Architecture Decision Records (ADRs)

Path: docs/architecture/

- `firebase-choice.md`: Why Firebase over SQLite
- `java-choice.md`: Why Java over Kotlin
- `github-actions.md`: Why GitHub CI/CD was used

13. Incident Response Playbook

File: `docs/incident-playbook.md`

markdown

CopyEdit

```
# 🚨 Incident Response Playbook

## Common Issues
- Firebase sync fails
- Crash on login

## Resolution Steps
- Check internet access
- Re-authenticate Firebase

## Contacts
- DevOps Owner: Nitya
- Firebase Admin: [Email or Contact]
```