ADDIS ABABA SCIENCE AND TECHNOLOGY UNIVERSITY

COLLEGE OF ELECTRICAL AND MECHANICAL ENGINEERING

DEPARTMENT OF SOFTWARE ENGINEERING

Course Name:Software Configuration Management

Section: B

Deliverable 2

Group Members

1. Dawit Sema	ETS 0455/14
2. Dawit Berhanu	ETS 0457/14
3. Delal Mohammad	ETS 0482/14
4. Eden Melaku	ETS 0496/14
5. Eyerusalem Kidane	ETS 0574/14
6. Gelila Nebiyu	ETS 0690/14
7. Hana Abiyu	ETS0728/14

Submitted to:Mr.Yibrah

Submission Date:30/12/2025

1. Configuration Item (CI) Identification

1.1 Purpose

This section identifies and documents all **Configuration Items (CIs)** for the **TO-DO List Manager** project.
A Configuration Item is any software artifact or document that is placed under formal configuration control and is subject to versioning, change control, and audit.

The purpose of CI identification is to:

- Ensure traceability of all project artifacts
- Control changes through approved SCM processes
- Support baseline creation, release management, and configuration audits

All CIs are managed using **Git and GitHub** and are uniquely identified, versioned, owned, categorized, and assigned a status throughout the project lifecycle.

1.2 CI Register

Configuration Item Register – TO-DO List Manager

CI ID	CI Name	Version	Owner	Category	Status
CI-01	Repository (todo-scm-project)	v1.1	Gelila	Repository	Active
CI-02	src/App.tsx	v1.1	Gelila	Source Code	Active
CI-03	src/components/Login.tsx	v1.1	Eden	Source Code	Active
CI-04	src/components/Dashboard.tsx	v1.1	Dawit	Source Code	Active
CI-05	src/components/* (other TSX files)	v1.1	Dawit S.	Source Code	Active

CI-06	data/ (localStorage / mock data)	v1.1	Gelila	Data	Active
CI-07	README.md	v1.1	Delal	Document	Approved
CI-08	LICENSE	v1.1	Hana	Document	Approved
CI-09	package-lock.json	v1.1	Eyerus	Configuration	Approved
CI-10	tailwind.config.js	v1.1	Eden	Configuration	Approved
CI-11	tsconfig.json	v1.1	Delal	Configuration	Approved
CI-12	.gitignore	v1.1	Hana	Configuration	Approved

1.3 CI Category Descriptions

- **Repository**
  The GitHub repository that stores and manages all project artifacts under version control.
- **Document**
  Planning and reporting artifacts such as the SCMP, README, CI Register, Change Requests, and audit reports.
- **Design**
  UI mockups and interface design artifacts used to guide implementation.
- **Source Code**
  React (TypeScript), HTML, CSS, and JavaScript files implementing the system's functionality.
- **Data**
  Simple data storage mechanisms such as browser localStorage or mock JSON data used to store tasks.
- **Configuration**
  Build, dependency, and environment configuration files required to run and maintain the project.

1.4 CI Control Mechanism

All Configuration Items listed in this register are subject to the following controls:

- Stored and maintained in the GitHub repository
- Versioned using Git commits and tags
- Modified only through approved Change Requests (CRs)
- Reviewed via Pull Requests before merging
- Included in baselines and releases
- Verified during Physical and Functional Configuration Audits

This ensures that no unauthorized or undocumented changes are introduced into the system.

1.5 Naming Conventions

Repository Naming

- Uses lowercase letters with hyphens for readability
- Format: `<project-topic>-scm-project`
- Example: `todo-scm-project`

This naming clearly reflects both the project domain and its academic SCM purpose.

Source Code File Naming

- React components use **PascalCase**, following React conventions:
  - `App.tsx`
  - `Login.tsx`
  - `Dashboard.tsx`
- File names are descriptive and reflect component responsibility.

Configuration File Naming

Standard industry-recognized names are preserved to avoid confusion:

- `tsconfig.json`
- `tailwind.config.js`
- `.gitignore`
- `package-lock.json`

Document Naming

Documents follow a clear versioned format:

DocumentName_Version.ext

Examples:

- `SCMP_v1.0.docx`
- `CI_Register_v1.1.xlsx`