IMPLEMENTATION

STRATEGY DOCUMENT

(ISD) FOR INCOME &

EXPENSE MANAGEMENT

STUDENT NAME: TIMOTHY KABURU

REGISTRATION NUMBER: 24/01455

UNIT CODE: DIT 503

UNIT TITLE: PROJECT FOR INCOME AND EXPENSE MANAGEMENT

SUPERVISOR: CLIVE ONSUMU

DATE SUBMITTED: 7/17/2025

**TABLE OF CONTENTS**

[**1. INTRODUCTION** 4](#_Toc13134)

[1.1 Purpose 4](#_Toc13135)

[1.2 Scope 4](#_Toc13136)

[1.3 Objectives 5](#_Toc13137)

[1.4 Audience 6](#_Toc13138)

[**2. IMPLEMENTATION APPROACHES** 7](#_Toc13139)

[2.1 Development Approach 7](#_Toc13140)

[2.2 Deployment Approach 8](#_Toc13141)

[2.3 Training Approach 8](#_Toc13142)

[**3. IMPLEMENTATION STRATEGY** 10](#_Toc13143)

[3.1 Project Overview 10](#_Toc13144)

[3.2 Project Phase 10](#_Toc13145)

[3.2.1 Planning Phase 10](#_Toc13146)

[3.2.2 Development Phase 11](#_Toc13147)

[3.2.3 Testing Phase 12](#_Toc13148)

[3.2.4 Deployment Phase 13](#_Toc13149)

[3.2.5 Maintenance Phase 13](#_Toc13150)

[3.3 Implementation Details 14](#_Toc13151)

[3.3.1 Development Environment 14](#_Toc13152)

[3.3.2 Deployment Architecture 15](#_Toc13153)

[3.3.3 Data Management 15](#_Toc13154)

[3.3.4 Legal and Compliance 15](#_Toc13155)

[4. RESOURCES 16](#_Toc13156)

[4.1 Team Roles 16](#_Toc13157)

[4.2 Tools 16](#_Toc13158)

[4.3 Budget 16](#_Toc13159)

[**5. IMPLEMENTATION SCHEDULE** 17](#_Toc13160)

[**6. RISK MANAGEMENT** 18](#_Toc13161)

[**7. ASSUMPTIONS** 18](#_Toc13162)

[**8. DELIVERABLES** 19](#_Toc13163)

[**9. LEGAL AND COMPLIANCE**  19](#_Toc13164)

[**CONSIDERATIONS** 19](#_Toc13165)

[**10. CONCLUSION** 19](#_Toc13166)

# INTRODUCTION

## Purpose

* This Implementation Strategy Document outlines the approach for developing, deploying, and maintaining the Income & Expense Manager
* A web-based front-end application designed to enable users to track income and expenses.
* The strategy ensures the solution is delivered on time, is user-friendly, and meets basic security and usability standards.

## Scope

• The implementation covers:

* Development-Building a single-page front-end application using

HTML, CSS, and JavaScript.

* Deployment- Hosting the application as a static website on platforms such as GitHub Pages or Netlify
* Testing- Validating functionality and usability in modern browsers.
* Training- Providing simple documentation and usage instructions.
* Maintenance- Updating the application as needed for browser compatibility or feature enhancements.

#### Exclusions

* No back-end server, database, or API integration.
* No SQL usage or persistent data storage beyond browser session.
* No third-party services beyond static file hosting.

## Objectives

* Deliver a fully functional, intuitive, and visually appealing income and expense tracking tool.
* Ensure the application is accessible on all modern browsers and devices.
* Minimize barriers to entry by making login simple and providing demo credentials.
* Clearly communicate to users the limitations of in-browser data storage.
* Provide thorough documentation and in-app guidance for self-sufficient use.
* Maintain the application for compatibility and address user feedback as needed.

## Audience

* Individual users seeking a simple financial tracking tool.
* Project managers and developers involved in front-end website delivery.
* Stakeholders interested in lightweight, static web tools.

# IMPLEMENTATION APPROACHES

## Development Approach

• Model- Waterfall (Single Phase) — Requirements, design, development, and testing are sequential and limited in scope.

#### Technology Stack

* HTML5, CSS3 (with custom styling and media queries for responsiveness)
* JavaScript ES6 (in-browser logic only)

#### User Authentication

• Simple in-browser credential check (hardcoded for demo purposes; no persistent user management).

#### Data Storage

• In-memory JavaScript array; data is lost when page is refreshed or closed.

## Deployment Approach

#### Static Hosting

* Deployment via GitHub Pages, Netlify or similar static hosting services.
* No server-side scripts or dynamic back-end required.

## Training Approach

#### README

* Open the application in your web browser (Chrome, Firefox, Edge, or

Safari).

* On the login screen, use the following demo credentials:
* Username: user
* Password: 1234
* Click the Login button.
* Once logged in, you'll see the dashboard.
* To add an entry:
* Enter a brief description (e.g., "Salary", "Groceries").
* Enter the amount in Kenyan Shillings (Ksh).
* Select whether it is Income or Expense.
* Click Add Entry.
* Th The dashboard displays three summary cards:
* Income: Total of all income entries
* Expense: Total of all expense entries
* Balance: Difference between income and expense
* These values update instantly as you add new entries. i.e. entry will appear in the History table below and update your summary.
* To log out, click the Logout button at the top right of the dashboard.
* You’ll be returned to the login screen.

#### Support

* Users may consult the README file or in-app documentation for troubleshooting.
* Optionally, project maintainers may provide an email address for feedback or questions.

# IMPLEMENTATION STRATEGY

## Project Overview

* The Income & Expense Manager will be implemented as a single HTML file containing embedded CSS and JavaScript.
* The project will move sequentially through planning, development, testing, deployment, and maintenance phases.
* The application will be published as a static website and supported with comprehensive documentation.

## Project Phase

### Planning Phase

• Duration: 1 day

#### Activities

* Finalize requirements and features.
* Design the user interface and user experience.
* Identify target browsers and devices for compatibility.

#### Deliverables

* Requirements summary.
* UI mockup (if needed).

### Development Phase

• Duration: 2 days

##### Activities

* Develop HTML structure.
* Implement CSS styling and responsive layout.
* Write JavaScript for login, entry management, summary calculations, and

UI updates.

##### Deliverables

* Complete HTML file.
* Embedded CSS and JavaScript.

### Testing Phase

• Duration:1 day

##### Activities

* Manual testing in Chrome, Firefox, Edge, and mobile browsers.
* Validate all features: login, adding entries, viewing summaries, logout.

Deliverables

* List of test cases and results.

### Deployment Phase

• Duration:1 day

##### Activities

* Upload HTML file to chosen static hosting platform.
* Verify accessibility and functionality.

Deliverables

* Live public URL for the application.

### Maintenance Phase

• Duration: Ongoing

##### Activities

* Fix browser compatibility issues.
* Update styles or features based on user feedback.

##### Deliverables

* Updated HTML file(s).
* Change logs and update notes.

## Implementation Details

### Development Environment

#### Tools

* Text Editor (VS Code, Sublime, Atom, etc.)
* Browser Developer Tools

### Deployment Architecture

* Front-End- Single static HTML file with embedded CSS and JavaScript.
* Hosting-Deployed to GitHub Pages or any static site host.
* Security- No data is stored permanently; all logic is client-side.

### Data Management

#### Storage

* All income/expense entries are stored in a JavaScript array in memory.
* Data is lost when the browser is closed or refreshed.
* No SQL or database integration.

### Legal and Compliance

* Data Privacy- No personal or sensitive data is stored or transmitted.
* Compliance- Application is compliant with privacy standards as no data leaves the user's browser.

# RESOURCES

## Team Roles

* Front-End Developer: Designs and builds the application.
* Tester: Validates functionality and browser compatibility.

## Tools

* Development: Text editor, browser.
* Testing: Browser-based manual testing.
* Deployment: GitHub Pages and Netlify.

## Budget

* Development: Minimal (developer time).
* Hosting: Free (using GitHub Pages or similar).
* Maintenance: None required unless updates are needed.

# IMPLEMENTATION SCHEDULE

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| PHASE | TASKS | DURATION | START DATE | END DATE |
| PLANNING | REQUIREMENTS,  DESIGN | 1 DAY | 2025/07/21 | 2025/07/21 |
| DEVELOPMENT | HTML/CSS/JS  CODING | 2 DAYS | 2025/07/22 | 2025/07/23 |
| TESTING | MANUAL BROWSER TESTING | 1 DAY | 2025/07/24 | 2025/07/24 |
| DEPLOYMENT | UPLOAD AND  PUBLISH SITE | 1 DAY | 2025/07/25 | 2025/07/25 |
| MAINTENANCE | UPDATES ARE  NEEDED | AS NEEDED | ONGOING | ONGOING |

# RISK MANAGEMENT

* Browser Compatibility Issues: Mitigation: Test on multiple browsers and devices.
* User Data Loss (refresh/close): Mitigation: Clearly inform users that data is not saved permanently.
* Limited Features: Mitigation: Scope is intentionally minimal for simplicity.

# ASSUMPTIONS

* Users understand that data is not stored persistently.
* No confidential or sensitive data is entered.
* Hosting platform supports static HTML/CSS/JS files.

# DELIVERABLES

* Final HTML file.
* Deployment on chosen host.
* Basic usage instructions.

# LEGAL AND COMPLIANCE

# CONSIDERATIONS

* No personal data storage or transmission.
* No external data processing.
* Application meets privacy expectations for static web tools.

# CONCLUSION

* This implementation strategy delivers a simple, browser-based Income & Expense Manager.
* The application is easy to use, requires no back-end, and can be deployed rapidly to any static hosting platform.
* Its front-end-only architecture ensures privacy and minimal maintenance.