



ACODES MUSHISHIRO TSS

Email: acodesmushishiro@gmail.com

Tel no: +250788412648



LEVEL THREE SOFTWARE DEVELOPMENT

FRONTENT ASSIGNMENT

MODULES COVERED:

- 1. WEBSITE DEVELOPMENT**
 - 2. JAVASCRIPT FUNDAMENTAL**
 - 3. CONDUCT VERSION CONTROL SYSTEM**
-
-

SIMS – Stock Inventory Management System (HTML • CSS • JavaScript)

As a frontend developer you are required to build a frontend **Stock Inventory Management System (SIMS)** that allows:

- User account creation (Register)
- User authentication (Login)
- Access-controlled dashboard
- Viewing and managing stock items

⚠ Only authorized (logged-in) users must be able to view and interact with the dashboard content.

No backend is required. All data must be handled using **JavaScript and localStorage**.

1. PAGES TO BE CREATED

The system must include the following pages or sections:

- 1. Register Page**
- 2. Login Page**
- 3. Dashboard Page (SIMS)**

SIMS
Stock Inventory Management System

John Doe
Administrator

Dashboard

Overview of your inventory status

Total Products	In Stock	Out of Stock
248 +12 this month	205 82.7% of total	43 17.3% of total

Products

Search products...

PRODUCT ID	PRODUCT NAME	CATEGORY	QUANTITY	PRICE	STATUS
PRO-001	Wireless Keyboard	Electronics	45	\$49.99	In Stock
PRO-002	USB-C Hub Adapter	Accessories	0	\$29.99	Out of Stock

Pages may be separate or dynamically displayed sections, but:

- Only one page is visible at a time

- Unauthorized users must never see dashboard elements

REQUIREMENTS FOR EACH PAGE

A. REGISTER PAGE

Required Elements

- Full Name input
- Email input
- Password input
- Confirm Password input
- Register button
- Validation message area

Validation Rules

- All fields are required
- Valid email format
- Email must be unique
- Password must:
 - Be at least 8 characters
 - Include uppercase, lowercase, and a number
- Password and Confirm Password must match

Expected Behavior

- User details saved in localStorage
- Duplicate registration blocked
- On success, redirect to Login page

B. LOGIN PAGE

Required Elements

- Email input
- Password input

- Login button
- Error message area

Validation Rules

- All fields required
- Email must exist
- Password must match registered user

Expected Behavior

- Successful login stores login state
- Redirect to Dashboard
- Failed login shows error message
- Page must not reload

C. DASHBOARD PAGE (SIMS)

 **Dashboard must be accessible ONLY after login**

Dashboard Elements

1. Header Area

- System name: **SIMS – Stock Inventory Management System**
- Logged-in user name
- Logout button

2. Sidebar Navigation

- Dashboard
- Products
- Reports (UI only)
- Settings (UI only)

Sidebar must NOT appear if the user is not logged in.

3. Summary Cards

Display cards showing:

- Total Products

- Products In Stock
- Out of Stock Products

Cards must be hidden from unauthorized users.

4. Products Table

Table Columns

- Product ID
- Product Name
- Category
- Quantity
- Price
- Status (In Stock / Out of Stock)

Behavior

- Products loaded dynamically using JavaScript
- Empty state shown if no products exist

5. Add New Product Button

- Positioned at the **top-right of the dashboard**
- Visible only to logged-in users
- Opens a form (modal or section)

Add Product Form Fields

- Product Name (required)
- Category (required)
- Quantity (number only)
- Price (number only)

Product Validation Rules

- No empty fields
- Quantity and price must be numbers
- Quantity cannot be negative

Expected Behavior

- Products saved in localStorage
- Products appear immediately in table
- Cards update dynamically

2. WHAT MUST BE DONE IN EACH TECHNOLOGY

HTML

- Use semantic HTML elements
- Proper heading structure
- All inputs must have <label>
- Use <table> correctly for products
- Use <section>, <aside>, <header> properly
- No inline JavaScript

CSS

- External CSS file only
- Responsive dashboard layout
- Use Flexbox and/or Grid
- Styled:
 - Sidebar navigation
 - Cards
 - Tables
 - Forms
- Visual states:
 - Hover
 - Error
 - Success
- Sidebar must collapse on small screens

JAVASCRIPT

- External JavaScript file only
- Handle:
 - User registration
 - User login
 - Authorization checks
 - Logout
- Control visibility of:
 - Sidebar
 - Cards
 - Product table
 - Add product button
- Use localStorage to store:
 - Users
 - Logged-in user
 - Products
- Dynamic DOM manipulation (no page reloads)
- Prevent dashboard access without login

SUBMISSION REQUIREMENT

All students **must follow these instructions** to submit their work properly.

1. Create a **new GitHub repository** using the following naming format:

Examples:

- ✓ group1_13soda
- ✓ group2_13sodb
- ✓ Add *readMe* file that holds group members

2. Clone the Repository Locally

3. Add Your Project Files

4. Commit & Push Your Work

5. Invite Trainers:

Invite **two trainers** to your repository:

fullstackv@proton.me

pattiradu@gmail.com

SIMS Project – 100 Marks Checklist

1. Website Development (HTML, CSS, JavaScript) – 60 Marks

Item	Marks	Check
Complete website structure (Register, Login, Dashboard with Products Table) using semantic HTML	10	<input type="checkbox"/>
Professional responsive design using external CSS (sidebar, cards, table, buttons styled properly)	15	<input type="checkbox"/>
Registration & Login fully functional with validation (email, strong password, prevent duplicates)	15	<input type="checkbox"/>
Dashboard functionality: Authorization, Logout, Summary Cards, Dynamic Products CRUD	10	<input type="checkbox"/>
Data persistence using localStorage + Proper error handling & validation	10	<input type="checkbox"/>

Subtotal: 60/60

2. Version Control System (Git & GitHub) – 40 Marks

Item	Marks	Check
Repository created correctly and project cloned locally	10	<input type="checkbox"/>
Proper Git workflow used (add, commit with clear messages, multiple commits)	15	<input type="checkbox"/>
Project pushed to GitHub and updated regularly	10	<input type="checkbox"/>
Trainers invited + Clean README documentation	5	<input type="checkbox"/>

Subtotal: 40/40

TOTAL: 100/100