**Candidate Programming Assignment – Web Application Mini Project - Intern**

**Duration**: 1 week  
**Purpose**: To assess your ability to design, build, and explain a working web application  
**Presentation**: You will present your work during the interview session (live demo + short explanation)

**🔧 Assignment Objective**

Build a **simple web application** that includes a product list management feature with user login, and select 2 from this 3 option, option 1) Time Stamp, option 2) Real-time Saving, option 3) End to end Encryption. You may use **any technology stack** you’re comfortable with (e.g., Node.js, PHP, Python, React, Laravel, Django, Firebase, etc.).

**📋 System Requirements**

**1. Login/Logout**

* Users can **log in** and **log out** securely.
* Session should persist while the user is active.

**2. Product Management**

* Track and display a list of products.
* Each product must include:
  + Product Code
  + Product Name
  + Description (long text box)
  + Price with **END-TO-END ENCRYPTION FUNCTION**
  + **Save** button
  + **Delete** button

**3. Time Stamp**

* When a product is saved (new add or edit), the system should record:
  + **Last edit date/time**
  + **Staff ID** (who edited)
  + **Last edited field name** (e.g., “Description”)

**4. Real-Time Save (RTS) Function**

* As users type/edit the product description or name:
  + Changes should auto-save **in the browser/session** (not in the main database).
  + This is to **prevent data loss** if the user navigates away or refreshes the page.
  + When returning, changes should still be there until the user clicks **Save**.

**5. End-to-end encryption function**

* With end-to-end encryption on the field, data or price stay between yourself and the people you choose, meaning not even server owner or maintenance team can see them.
  + Showing garbled characters in database.

**✅ Completion Expectations**

* Basic UI (no need to focus heavily on design, but must be usable)
* User-friendly experience and visible role-based access restrictions
* Working RTS for form fields (use localStorage, sessionStorage, or similar)
* Code should be structured and readable (comments are a plus)
* Be prepared to do a **5–10 min walkthrough** during the interview (what you built, how you built it, challenges faced)

**💡 Tips**

* Focus on **functionality and clarity**, not perfection
* You may use **AI tools**, documentation, and public libraries to assist you (mention them if used)
* Keep it simple but functional – this is not about building the perfect app, but showing your **problem-solving and coding ability**