Απαντήσεις στις ερωτήσεις της Δομής του Project Report

**1. Εισαγωγή & Motivation**

* **Τι αφορά το project;**Το project αφορά τη διαχείριση της αποθήκης ενός σούπερ μάρκετ μέσω ενός συστήματος που επιτρέπει την προσθήκη, ενημέρωση και αποστολή προϊόντων. Επίσης, υπάρχει σύνδεση με Arduino για την επικοινωνία με εξωτερικές συσκευές.
* **Ποιο πρόβλημα λύνει;**Το σύστημα λύνει το πρόβλημα της οργανωμένης διαχείρισης αποθέματος, αποτρέποντας ελλείψεις ή υπερβολική συσσώρευση προϊόντων. Επιπλέον, αυτοματοποιεί την επικοινωνία με το Arduino για την αποστολή ειδοποιήσεων ή αυτοματισμών.
* **Γιατί επιλέξατε αυτό το αντικείμενο;**Επιλέξαμε αυτό το project διότι είναι εύχρηστο , εύκολο στην κατανόηση και βασίζεται σε συστήματα που χρησιμοποιούμε στην καθημερινότητα.

**2. Objective & Scope**

* **Ποιοι είναι οι βασικοί στόχοι;**
  1. Ανάπτυξη ενός αποδοτικού συστήματος διαχείρισης αποθήκης.
  2. Δυνατότητα αποθήκευσης και ανάκτησης πληροφοριών προϊόντων.
  3. Αυτοματοποιημένη ενημέρωση για την κατάσταση των προϊόντων.
  4. Επικοινωνία με Arduino για ειδοποιήσεις επιτυχούς ή αποτυχημένης αποστολής προϊόντων.
* **Ποιες δυνατότητες περιλαμβάνονται;**
  1. Προσθήκη & Ενημέρωση προϊόντων στη βάση δεδομένων.
  2. Προβολή αποθέματος με οργανωμένο πίνακα.
  3. Διαχείριση αποστολής προϊόντων με αυτόματη ενημέρωση του αποθέματος.
  4. Επικοινωνία με Arduino για επιβεβαίωση ή σφάλμα κατά την αποστολή προϊόντων.
  5. Αποθήκευση και φόρτωση δεδομένων μέσω αρχείου warehouse.txt.

**3. System Architecture**

* **Πώς λειτουργεί το σύστημα;**Το σύστημα λειτουργεί με δομή βασισμένη σε αντικείμενα (OOP), όπου κάθε προϊόν αναπαρίσταται ως αντικείμενο της κλάσης Product. Τα προϊόντα αποθηκεύονται σε έναν vector, ενώ η ανάγνωση και εγγραφή δεδομένων γίνεται μέσω αρχείου warehouse.txt.
* **Ποιες είναι οι βασικές του συνιστώσες;**
  + Κλάση Product: Αποθηκεύει πληροφορίες προϊόντων (barcode, όνομα, ποσότητα, τιμή).
  + Vector warehouse: Διατηρεί τα προϊόντα που υπάρχουν στην αποθήκη.
  + Συναρτήσεις Διαχείρισης:
    - addOrUpdateProduct(): Προσθήκη ή ενημέρωση προϊόντος.
    - showProducts(): Προβολή των διαθέσιμων προϊόντων.
    - shipProduct(): Αποστολή προϊόντος και ενημέρωση του Arduino.
  + Arduino Communication: Αποστολή σημάτων επιτυχίας ή σφάλματος μέσω της sendToArduino().
* Διάγραμμα ροής

**4. Τεχνολογίες που Χρησιμοποιήθηκαν**

* **Ποιες και γιατί;**
  + C++: Γρήγορη και αποδοτική γλώσσα προγραμματισμού για τη διαχείριση δεδομένων.
  + Windows API (<windows.h>): Χρησιμοποιείται για την επικοινωνία με το Arduino μέσω σειριακής θύρας.
  + Arduino: Παρέχει hardware επικοινωνία και αυτοματοποιεί την επεξεργασία παραγγελιών.
* **Αναφορά βιβλιοθηκών που χρησιμοποιήθηκαν**
  + <iostream>: Διαχείριση εισόδου/εξόδου.
  + <fstream>: Ανάγνωση και εγγραφή σε αρχεία.
  + <vector>: Διαχείριση δυναμικών δομών δεδομένων.
  + <ctime>: Χρονομέτρηση της λειτουργίας του προγράμματος.
  + <windows.h>: Επικοινωνία με Arduino μέσω σειριακής θύρας.

**5. Κώδικας & Υλοποίηση**

* **Πώς υλοποιήθηκε η βασική λειτουργικότητα;**  
  Ο κώδικας ακολουθεί αρχές Αντικειμενοστραφούς Προγραμματισμού (OOP), διαχειρίζεται δυναμικά τα προϊόντα μέσω vector και χρησιμοποιεί το Windows API για σειριακή επικοινωνία με το Arduino.
* **Υπάρχουν σημαντικοί αλγόριθμοι ή δομές δεδομένων;**
  + nεπεξεργασία προϊόντων.
  + Η σειριακή επικοινωνία με το Arduino βασίζεται σε DCB (Device Control Block) και WriteFile().
  + Ο πίνακας showProducts() χρησιμοποιεί setw() για σωστή στοίχιση των δεδομένων.
* Ανεβάστε τον κώδικά σας σε ένα repo και γράψτε οδηγίες εκτέλεσης.

**6. Αποτελέσματα & Demo**

* Εμφανίζονται σωστά τα προϊόντα στην αποθήκη.
* Γίνεται σωστή επεξεργασία αποθεμάτων (προσθήκη, ενημέρωση, αποστολή).
* Η επικοινωνία με το Arduino είναι επιτυχής, στέλνοντας επιβεβαίωση ή σφάλμα ανάλογα με την κατάσταση της παραγγελίας.
* Το σύστημα μπορεί να αποθηκεύει και να φορτώνει δεδομένα μέσω του warehouse.txt.

**7. Comparison with AI Generated Code**

* **Πλεονεκτήματα του χειροκίνητου κώδικα:**
  + Καλύτερη προσαρμογή στις ανάγκες του project.
  + Ευελιξία και δυνατότητα επέκτασης.
  + Καλύτερη διαχείριση μνήμης και επιδόσεων.
* **Πιθανά πλεονεκτήματα του AI-generated κώδικα:**
  + Ταχύτερη ανάπτυξη αρχικού κώδικα.
  + Μείωση τυπογραφικών λαθών.
  + Αυτόματη βελτιστοποίηση ορισμένων τμημάτων.

**8. Conclusions & Lessons Learned**

* Η χρήση του OOP επέτρεψε εύκολη διαχείριση των προϊόντων και μελλοντική επεκτασιμότητα.
* Η σειριακή επικοινωνία με το Arduino βελτίωσε την αυτοματοποίηση του συστήματος.
* Η ορθή διαχείριση αρχείων (warehouse.txt) ήταν κρίσιμη για τη διατήρηση των δεδομένων.
* Το πρόγραμμα θα μπορούσε να επεκταθεί με γραφικό περιβάλλον (GUI) για ευκολότερη χρήση.
* Ενδεχομένως, θα μπορούσε να υλοποιηθεί πρόσβαση μέσω δικτύου για online διαχείριση της αποθήκης.