



CENG 3507 : Web Development Final Project : Blueberry Factory Management System

Project Overview:

This project will create a web-based management system for a factory that buys blueberries from farmers, categorizes and packages them, and sells them to customers. The system will enable the factory to track blueberry purchases, categorize blueberries into various package sizes with distinct pricing, and record sales transactions. The application will also provide detailed financial calculations, including income, expenses, taxes, and inventory tracking.

Project Objectives:

- 1. Manage Supplier Relations:** Track purchases from farmers, including price per kilogram and total purchase quantity.
- 2. Blueberry Packaging:** Sort and categorize blueberries into seven distinct packaging categories, each with different weight specifications and pricing.
- 3. Sales Management:** Process orders from customers, including recording the quantity sold, unit price, and total sales for each category.
- 4. Financial Calculations:** Calculate total revenue, expenses, net income, and tax based on sales and purchasing data.
- 5. Inventory Management:** Track the quantity of blueberries in each category to optimize stock availability and meet demand.
- 6. Comprehensive Reporting:** Provide an overview report including total income, expenses, tax calculation, number of products sold per category, and overall inventory remaining.

Project Features:

1. Supplier Management Module

- *Farmers' Information* : Store details such as farmer ID, name, contact, and location.
- *Purchase Records* : Record details of each purchase, including date, quantity, price per kilogram, and total cost.
- *Expense Calculation*: Calculate total expenses based on purchases to determine the cost of raw materials.

1.1. Farmers' Information

- Data Fields:
 - Farmer ID (unique identifier)
 - Name
 - Contact details (phone, email, etc.)
 - Location (address, region, GPS coordinates)
- Functionalities:
 - Add, update, and delete farmer profiles.

- Search and filter farmers by location or name.
- Export farmer details for reporting.

1.2. Purchase Records

- Data Fields:
 - Purchase ID (unique identifier)
 - Farmer ID (linked to Farmers' Information)
 - Date of purchase
 - Quantity purchased (e.g., in kilograms)
 - Price per kilogram
 - Total cost (calculated as $\text{Quantity} \times \text{Price per kilogram}$)
- Functionalities:
 - Log new purchases with automated calculations for total cost.
 - View and sort records by date, farmer, or amount.
 - Generate purchase summaries for individual farmers or time periods.

1.3. Expense Calculation

- Data Fields:
 - Total cost (summed from Purchase Records)
 - Time period (e.g., daily, weekly, monthly)
- Functionalities:
 - Automatically calculate total expenses for selected periods.
 - Generate reports on raw material costs.

2. Product Categorization and Packaging Module

2.1. Product Categories

- Category 1: Small (100g)
- Category 2: Medium (250g)
- Category 3: Large (500g)
- Category 4: Extra Large (1kg)
- Category 5: Family Pack (2kg)
- Category 6: Bulk Pack (5kg)
- Category 7: Premium (custom weight for special orders)
- Functionalities:
 - Assign products (blueberries) to specific categories based on their weight.
 - Ensure each product is categorized correctly to streamline packaging and shipping.
 - Track products by category to prevent confusion and improve logistics.

2.2. Pricing Structure

- Data Fields:
 - Category Price: Define a unique price for each category based on:
 - Weight (larger packs may have a reduced price per kilogram).
 - Market demand (price can fluctuate based on demand or seasonality).
- Functionalities:
 - Set and update prices based on market trends and weight.
 - Display pricing structure for each category.

- Calculate total cost for customers based on the category and quantity ordered.

2.3. Inventory Tracking

- Data Fields:
 - Stock Level: Track the quantity of blueberries in each category.
 - Restock Alerts: Set minimum stock thresholds to trigger alerts for restocking.
- Functionalities:
 - Monitor the stock of blueberries by category.
 - Alert when stock levels are low to prevent stockouts.
 - Generate reports on inventory usage, trends, and predictions for future stock needs.

3. Sales Management Module

Order Management:

- Order ID: Unique identifier for each order.
- Customer Details: Name, contact, and shipping information.
- Product Category: The category of the product ordered (e.g., Small, Medium, Large, etc.).
- Quantity Ordered: Number of units the customer ordered.
- Total Price: Total cost of the order, calculated as $\text{Quantity} \times \text{Unit Price}$.
- Order Status: Status of the order (e.g., Pending, Processed, Shipped, Delivered).

Functionalities:

- Log new customer orders.
- Update order status throughout the sales process.
- Search and filter orders by status, customer, or product category.
- Display order history and details for customers.

Revenue Calculation:

- Revenue: Total revenue from a particular order or group of orders.
- Product Category: The category of the products ordered.
- Unit Price: Price per unit of the product.
- Quantity Sold: Number of units sold.

Functionalities:

- Automatically calculate revenue based on the total price of all orders.
- Track and update revenue for each product category.
- Provide insights into revenue generated from each category.

Sales Reports:

- Units Sold: The number of units sold for each product category.
- Revenue Per Category: Revenue generated from each product category.
- Overall Revenue: Total revenue across all categories.

Functionalities:

- Generate detailed sales reports by product category, customer, or time period.
- Provide visual reports such as bar charts or pie charts to show sales trends.
- Export reports in formats like CSV or PDF for sharing and analysis.

4. Financial Analysis Module

- Income and Expense Calculation: Calculate overall income from sales and expenses from blueberry purchases.
- Tax Calculation: Apply tax rates to income for accurate financial reporting.
- Net Profit Calculation : Calculate net profit by deducting expenses and taxes from total revenue.

5. Inventory Management Module

To monitor and manage the stock levels of blueberries across different categories, ensuring optimal availability to meet demand.

Data Fields:

- Item ID: Unique identifier for each blueberry category (e.g., Fresh, Frozen, Organic).
- Category: Type of blueberries (e.g., Fresh, Frozen, Organic).
- Quantity Available: Current stock level (e.g., in kilograms).
- Reorder Level: Minimum quantity threshold to trigger restocking.
- Restock Date: Scheduled or actual date for restocking.
- Storage Location: Details of where the stock is stored (e.g., warehouse section, bin number).

Functionalities:

1. Real-Time Tracking:

- Update stock levels after sales or purchases.
- View stock levels by category.

2. Demand Forecasting:

- Analyze sales trends to predict future demand.
- Provide recommendations for stock adjustments.

3. Alerts and Notifications:

- Notify when stock falls below the reorder level.
- Send reminders for scheduled restocking.

4. Reporting:

- Generate inventory summaries (daily, weekly, monthly).
- Analyze turnover rates for each category.

5. Integration:

- Link with the Purchase Records module to adjust inventory upon receiving new stock.
- Connect to the Supplier Management module to streamline restocking.

6. Comprehensive Report Generation Module

- Generate an end-of-period report including:
 - Total income from sales
 - Total expenses from purchases
 - Tax applied
 - Net profit
- Number of products sold per category
- Remaining stock per category

Final Deliverables:

1. Fully Functional Web Application: Complete with all modules.
2. User Manual: Documentation for users, detailing how to navigate the application.
3. Technical Report: Explanation of code structure, functionalities, and calculations.
4. Financial Report Samples: Generated example reports based on sample data.
5. Presentation: Summary presentation showcasing project features, calculations, and final results.

Example Use-Cases :

Importantly : Use-Cases scenarios jus for examples

Use Case 1: Purchase Blueberries from Farmers

Objective: Track purchases and record expenses.

Scenario:

1. Actor: Factory Manager
2. Trigger: The factory receives a shipment of blueberries from a farmer.
3. Preconditions: Farmer details are in the system, and the manager is logged in.
4. Steps:
 - Manager opens the “Supplier Management” module.
 - Selects the farmer from a list or adds a new farmer.
 - Enters details of the purchase: quantity (in kilograms) and price per kilogram.
 - The system calculates the total purchase cost and saves it as an expense.
5. Postconditions:
 - The total expense is updated in the system.
 - The inventory is updated with the new blueberry stock.
6. Expected Outcome: The purchase is recorded, the farmer’s details are updated, and expenses are accurately reflected in the system.

Use Case 2: Categorize and Package Blueberries

Objective: Sort blueberries into seven packaging categories with assigned prices.

Scenario:

1. Actor: Inventory Staff
2. Trigger: The factory has processed and cleaned the blueberries, readying them for packaging.

3. Preconditions: Inventory includes sufficient blueberries, and the staff is logged in.
4. Steps:
 - Staff member opens the “Product Categorization” module.
 - Selects the category type (e.g., Small, Medium, Family Pack) and enters the quantity of blueberries to package in each.
 - System updates inventory, deducting the blueberries used for packaging.
 - Pricing is applied automatically based on the category selected.
5. Postconditions:
 - Blueberries are categorized and ready for sale.
 - Inventory count is updated per category.
6. Expected Outcome: Blueberries are accurately categorized and packaged, with inventory adjusted accordingly.

Use Case 3: Record and Process Sales

Objective: Track customer orders, calculate revenue, and update inventory.

Scenario:

1. Actor: Sales Staff
2. Trigger: A customer places an order for packaged blueberries.
3. Preconditions: The product categories are stocked and priced.
4. Steps:
 - Sales staff logs into the “Sales Management” module.
 - Enters the customer’s order details, selecting the category and quantity for each package type.
 - System calculates total order cost based on category prices and quantity.
 - Inventory count is updated to reflect the sale.
5. Postconditions:
 - Sale is recorded in the system.
 - Revenue is updated in the financial reports.
 - Inventory is adjusted to reflect the sold packages.
6. Expected Outcome: Customer order is processed, income is recorded, and inventory is updated.

Use Case 4: Generate Financial and Inventory Report

Objective: Produce an end-of-period report detailing income, expenses, taxes, and inventory status.

Scenario:

1. Actor: Factory Manager
2. Trigger: End of the month or quarter, requiring a financial report.
3. Preconditions: Sales and purchase records are up-to-date.
4. Steps:
 - Manager opens the “Report Generation” module.
 - Selects the desired reporting period (e.g., monthly or quarterly).
 - System generates a report detailing:
 - Total income from sales
 - Total expenses from blueberry purchases
 - Tax calculated on revenue
 - Net profit (income - expenses - tax)
 - Number of products sold per category
 - Current stock for each product category
5. Postconditions:

- Report is available in downloadable format (e.g., PDF or CSV).
 - Manager reviews financial performance and remaining inventory.
6. Expected Outcome: Accurate and comprehensive financial report generated for the specified period.

Use Case 5: Inventory Reorder Alert

Objective: Notify staff when stock of a category runs low.

Scenario:

1. Actor: Inventory Staff
2. Trigger: Packaging process checks inventory levels.
3. Preconditions: System has minimum stock thresholds set for each category.
4. Steps:
 - Staff initiates packaging process in the “Product Categorization” module.
 - System checks current inventory against the minimum threshold.
 - If inventory for a category is below the threshold, an alert is triggered.
5. Postconditions:
 - Staff is notified that stock is running low.
 - System suggests reordering from suppliers.
6. ****Expected Outcome****: Low-stock alert prompts staff to reorder blueberries to maintain sufficient inventory.

Use Case 6: Tax Calculation and Verification

Objective: Calculate tax based on revenue for financial reporting.

Scenario:

1. Actor: Accountant
2. Trigger: End-of-period tax report required for financial analysis.
3. Preconditions: Income and expense records are finalized for the period.
4. Steps:
 - Accountant accesses the “Financial Analysis” module.
 - Selects tax calculation, based on revenue and applicable tax rate.
 - System calculates the tax and applies it to the final report.
5. Postconditions:
 - Tax liability is recorded.
 - Tax amount is deducted from revenue to update net profit.
6. Expected Outcome: Accurate tax calculation to comply with financial reporting standards.



ChatGPT allowed:

You can use chatGPT just for search not copy-paste!!

Not : The report can be a simple one-page(html page) text document. There's no need to generate a PDF. You have flexibility in the design and structure; the evaluation will focus solely on your understanding and application of HTML, CSS, and JavaScript.