**BTEC Assignment Brief**

|  |  |
| --- | --- |
| **Qualification** | Pearson BTEC Higher Nationals in Digital Technologies |
| **Unit number and title** | Unit 4: Programming |
| **Learning aim(s)** | By the end of this unit, students will be able to:  **LO1** Define basic algorithms to carry out an operation and outline the process of programming an application.  **LO2** Explain the characteristics of procedural, object-orientated, and event-driven programming.  **LO3** Implement basic algorithms in code using an IDE.  **LO4** Determine the debugging process and explain the importance of a coding standard. |
| **Assignment title** | **Kindergarten Meal Tracking & Inventory Management System** |
| **Assessor** | Shukurov Sanjar |
| **Issue date** |  |
| **Hand in deadline** |  |

|  |  |
| --- | --- |
| **Scenario or Context** | **Kindergarten Meal Tracking & Inventory Management System** **Project Overview:** You are tasked with developing a full-stack application for a kindergarten that helps staff manage kitchen inventory, meal tracking, and report generation. The application must be robust, reflect real-world operations, and include features to monitor ingredient usage, portions served, and potential misuse of ingredients. **Functional Requirements:******1. Product Management (Ingredients)****  * Staff must be able to add new products (e.g., beef, potato, salt) and their current total weight in grams. * Staff must be able to update or delete product quantities. * Add delivery date to track when new ingredients arrive.  ****2. Meal Management****  * Staff should be able to define meals by name. * Each meal consists of specific quantities of ingredients. * Ability to update or delete meals and their recipes.  ****3. Meal Serving System****  * A button labelled **"Serve Meal"** must deduct the required amount of ingredients from inventory. * If any ingredient is insufficient, serving must be disallowed with an alert. * The system must log the date/time of the meal, and the user responsible for serving it.  ****4. Portion Estimation****  * Based on current inventory, estimate and display how many full portions of each meal can be made. * This should dynamically update whenever inventory or meals change.  ****5. Visualization / Reports****  * Graph of ingredient usage over time, including both consumption and delivery. * Monthly summary graph showing:   + Total number of portions served.   + Total number of portions that could have been served.   + Difference rate (%). If the rate exceeds 10–15%, flag potential misuse.  ****6. User Tracking****  * Log and display which user (staff member) served which meal. * Display a table of all served meals with staff name and timestamp.  ****7. Role-Based Permissions****  * Admins: Full access, including reports and settings. * Cooks: Can only serve meals. * Managers: Can update inventory and view analytics.  ****8. Notifications / Alerts****  * Warn when an ingredient’s quantity falls below a threshold. * Alert if monthly discrepancy rate exceeds 15%.  ****9. Background Tasks****  * Use Celery (or similar) to generate monthly reports or recalculate estimates.  ****10. Real-Time Updates (WebSockets)****  * Live display of inventory updates and ingredient levels. * Real-time warning if an ingredient is running out during meal serving.  **Technology Suggestions**  * **Backend**: Django (DRF) or FastAPI * **Frontend**: React, Vue.js, or vanilla HTML/CSS with Chart.js * **Database**: PostgreSQL preferred; SQLite allowed for simplicity * **Task Queue**: Celery with Redis (for bonus background tasks) * **Charts**: Chart.js or Recharts * **Real-time**: WebSockets (Django Channels / FastAPI with WebSocket routes)  **Assignment Timeline (Suggested)******Stage 1****: Ingredient and Meal CRUD (basic forms, DB models)****Stage 2****: Implement Meal Serving Logic (with deduction + logging)****Stage 3****: Add Portion Estimation Algorithm****Stage 4****: Data Visualization & Monthly Reports****Bonus Stage****: Roles, Alerts, WebSocket Integration**Deliverables**  * Full source code (backend + frontend if applicable) * Database schema * Sample data (fixtures) * Screenshots of key pages * README with setup instructions  **Evaluation Criteria**  * Functional correctness (deduction, portion calc, graphs) * Code organization and clarity * Use of appropriate tools and libraries * Creativity and robustness (bonus features) * Data integrity and security (role-based access, validation) |
|  |  |

Learning Outcomes and Assessment Criteria

|  |  |  |
| --- | --- | --- |
| **Pass** | **Merit** | **Distinction** |
| **LO1** Define basic algorithms to carry out an operation  and outline the process of programming an application | | **D1** Evaluate the  implementation of an  algorithm in a suitable  language and the relationship  between the written  algorithm and the code  variant. |
| **P1** Provide a definition of  what an algorithm is and  outline the process in  building an application. | **M1** Determine the steps  taken from writing code to  execution. |
| **LO2** Explain the characteristics of procedural, object-  orientated and event-driven programming | | **D2** Critically evaluate the source code of an application which implements the procedural, object-orientated and event driven paradigms, in terms of the code structure and characteristics. |
| **P2** Give explanations of what procedural, object-orientated and event-driven paradigms are; their characteristics and the relationship between them. | **M2** Compare and contrast  the procedural, object orientated and event driven paradigms used in given source code of an application |
| **LO3** Implement basic algorithms in code using an IDE | | **D3** Evaluate the use of an IDE for development of applications contrasted with not using an IDE. |
| **P3** Write a program that implements an algorithm using an IDE. | **M3** Use the IDE to manage the development process of the program. |
| **LO4** Determine the debugging process and explain the importance of a coding standard | | **D4** Critically evaluate why a coding standard is necessary in a team as well as for the individual. |
| **P4** Explain the debugging process and debugging facilities available in the IDE. | **M4** Evaluate how the debugging process can be used to help develop more secure, robust applications. |
| **P5** Outline the coding standard you have used in your code. |

|  |  |
| --- | --- |
| **Sources of information to support you with this Assignment** | **Submission Requirements:**   * Written reports for P1, P2, P4, P5, M1, M2, M3, M4, D1, D2, D3, D4. * Flowcharts or diagrams for M1. * Source code files for the Kindergarten Meal Tracking & Inventory Management System. * A development journal documenting the challenges, learning outcomes, and reflections on the project.   **Books:**  Dennis, A. and Haley, W. (2009) *Systems Analysis and Design*. John Wiley & Sons Ltd.  Ferguson, J. (2014) *BDD in Action: Behavior-driven development for the whole software lifecycle*. Manning.  Lejk, M. and Deeks, D. (2002) *An Introduction to System Analysis Techniques*. 2nd Ed. Addison-Wesley.  Murch, R. (2012) *The Software Development Lifecycle: A Complete Guide*. Kindle. |