**Department of Collegiate & Technical Education**

**Capstone Project**

**Format-3**

**Time Line Structure**

**Capstone Project Name:** INVENTORY MANAGEMENT SYSTEM

**Capstone Project Members:** SHESHAGIRI S [149CS21046]

DIVJA J [149CS21021]

DILEEP G P [149CS21020]

DHANYASHREE A P [149CS21019]

**1} Identify the activities and tasks needed to produce each work package.**

1. System Requirements:

* Activities:
  + Gather user requirements through interviews, surveys, or workshops.
  + Analyze requirements to identify functional and non-functional needs.
  + Define user roles and permissions for access control.
  + Document requirements in detail with specifications and use cases.
* Tasks:
  + Conduct user interviews and surveys.
  + Define system functionalities and features.
  + Choose appropriate technology stack (e.g., PHP framework, database).
  + Specify performance and security requirements.
  + Create user roles and permission matrix.
  + Document requirements in a formal document.

2. Database Design:

* Activities:
  + Design database schema with tables, relationships, and data types.
  + Choose appropriate database technology (e.g., MySQL, PostgreSQL).
  + Implement database connection and interaction in PHP.
* Tasks:
  + Define entity-relationship diagram (ERD) for data model.
  + Design tables for products, categories, suppliers, orders, etc.
  + Define data types for each field (e.g., string, integer, date).
  + Establish relationships between tables (e.g., foreign keys).
  + Choose and install database software.
  + Write PHP code for database connection and queries.

3. Product Management Module:

* Activities:
  + Develop functionalities for adding, editing, and deleting products.
  + Implement product image upload and management.
  + Design product category and subcategory structure.
  + Track product stock levels and reorder points.
* Tasks:
  + Create forms for product data entry and editing.
  + Implement data validation and error handling.
  + Develop product image upload and storage logic.
  + Design category and subcategory hierarchy with selection options.
  + Implement functions for stock level tracking and reorder point calculations.
  + Write PHP code for database interactions with product data.

4. Purchase Management Module:

* Activities:
  + Develop functionalities for creating purchase orders for suppliers.
  + Track purchase order status and receive products.
  + Manage supplier information and contracts.
  + Calculate and record purchase costs.
* Tasks:
  + Design purchase order form with product selection and quantity options.
  + Implement purchase order approval workflow.
  + Develop supplier management interface with contact details and contracts.
  + Integrate with supplier APIs (if applicable) for product data.
  + Write PHP code for purchase order creation, tracking, and cost calculations.

5. Sales Management Module:

* Activities:
  + Process customer orders with product selection and quantity.
  + Manage sales invoices and receipts with payment processing.
  + Apply discounts and promotions to orders.
  + Track sales performance and generate reports.
* Tasks:
  + Create shopping cart or order form for customer product selection.
  + Implement customer authentication and order placement logic.
  + Integrate with payment gateway for secure transactions.
  + Design discount and promotion rules and application.
  + Develop reports for sales by product, customer, and period.
  + Write PHP code for order processing, payment handling, and reporting.

6. Reporting Module:

* Activities:
  + Generate inventory reports (stock levels, reorder points, etc.).
  + Generate sales reports (sales by product, by customer, etc.).
  + Generate purchase reports (purchase costs, supplier performance, etc.).
* Tasks:
  + Design report templates with data visualization options.
  + Implement report generation logic based on user-defined criteria.
  + Allow export of reports to various formats (PDF, CSV, etc.).
  + Write PHP code for report data retrieval and formatting.

7. Security Implementation:

* Activities:
  + Implement user authentication and authorization mechanisms.
  + Encrypt sensitive data (e.g., passwords).
  + Protect against common vulnerabilities (e.g., SQL injection).
  + Secure data transmission and storage.
* Tasks:
  + Implement user login and registration features.
  + Use secure password hashing and storage techniques.
  + Validate user input and sanitize data to prevent attacks.
  + Implement secure communication protocols (e.g., HTTPS).
  + Encrypt database data and backups.

8. User Interface Design & Development:

* Activities:
  + Design user-friendly interface prototypes for each module.
  + Develop core UI components with responsive design.
  + Ensure accessibility for users with

**2} Identify resources for each task (e.g., time, knowledge, monetary costs etc.)**

1. System Requirements:

* Time: 1-2 weeks (gathering, analysis, documentation)
* Knowledge: Business analysis, user experience design
* Monetary Costs: May involve user research tools/consultants

2. Database Design:

* Time: 1-2 weeks (schema design, implementation)
* Knowledge: Database design principles, chosen database technology
* Monetary Costs: Database software licensing (if applicable)

3. Product Management Module:

* Time: 2-4 weeks (core functionalities, image handling, categories)
* Knowledge: PHP development, user interface design, data validation
* Monetary Costs: Image storage depending on volume (cloud storage etc.)

4. Purchase Management Module:

* Time: 2-3 weeks (purchase orders, supplier management, cost calculations)
* Knowledge: PHP development, API integration (if applicable), payment gateway integration (if applicable)
* Monetary Costs: API integration fees (if applicable), payment gateway fees (if applicable)

5. Sales Management Module:

* Time: 3-4 weeks (order processing, payment handling, discounts, reports)
* Knowledge: PHP development, user authentication, security, payment gateway integration, reporting tools
* Monetary Costs: Payment gateway fees (if applicable), reporting tool licenses (if applicable)

6. Reporting Module:

* Time: 1-2 weeks (data retrieval, report generation, formatting)
* Knowledge: PHP development, reporting tools/libraries, data visualization
* Monetary Costs: Reporting tool licenses (if applicable)

7. Security Implementation:

* Time: 2-3 weeks (authentication, data encryption, vulnerability protection)
* Knowledge: Security best practices, secure coding techniques, chosen security libraries
* Monetary Costs: Security library licenses (if applicable)

8. User Interface Design & Development:

* Time: 2-3 weeks (prototyping, core UI, responsive design)
* Knowledge: User interface design principles, front-end development (HTML, CSS, JavaScript)
* Monetary Costs: Design tools licenses (if applicable)

**3} Estimate how long it will take to complete each task. Consider constraints - resources, time, knowledge**

1. System Requirements:

* Time: 2-3 weeks (gathering & analysis) + 1-2 weeks (documentation) = 3-5 weeks
  + User research workshops or surveys might extend the gathering & analysis phase.
  + Business analyst or UX designer involvement can expedite documentation.

2. Database Design:

* Time: 1-2 weeks (schema design) + 1-2 weeks (implementation) = 2-4 weeks
  + Complex data models with intricate relationships might require more design time.
  + Database administrators or experienced developers can streamline implementation.

3. Product Management Module:

* Time: 2-3 weeks (core functions) + 1-2 weeks (image handling) + 1 week (categories) = 4-6 weeks
  + API integration for product data retrieval can add complexity and time.
  + Strong understanding of UI/UX design and data validation is crucial.

4. Purchase Management Module:

* Time: 2-3 weeks (purchase orders) + 1-2 weeks (supplier management) + 1-2 weeks (cost calculations) = 4-7 weeks
  + Third-party API integrations and payment gateway setup add complexity.
  + Knowledge of API integrations and secure payment processing is essential.

5. Sales Management Module:

* Time: 2-3 weeks (order processing) + 1-2 weeks (payment handling) + 1-2 weeks (discounts) + 1-2 weeks (reports) = 6-9 weeks
  + Integrating with complex payment gateways and generating detailed reports require more effort.
  + Expertise in secure payment processing and reporting tools is necessary.

6. Reporting Module:

* Time: 1-2 weeks (data retrieval) + 1-2 weeks (report generation) + 0.5-1 week (formatting) = 2.5-5 weeks
  + Designing advanced reports with data visualization elements takes more time.
  + Familiarity with reporting tools and data visualization techniques is beneficial.

7. Security Implementation:

* Time: 1-2 weeks (authentication) + 1-2 weeks (data encryption) + 1-2 weeks (vulnerability protection) = 3-6 weeks
  + Implementing multi-factor authentication or advanced encryption adds complexity.
  + Deep understanding of secure coding practices and penetration testing methodologies is essential.

8. User Interface Design & Development:

* Time: 1-2 weeks (prototyping) + 1-2 weeks (core UI) + 1-2 weeks (responsive design) = 3-6 weeks
  + Implementing visually complex or interactive UI elements takes more time.
  + Expertise in front-end frameworks and responsive design principles is crucial.

**4} Determine which tasks are dependent on other tasks and develop a critical path.**

**Dependencies:**

1. System Requirements must be completed before Database Design can start, as the database structure needs to align with the functionality requirements.
2. Database Design must be completed before Product Management Module, Purchase Management Module, Sales Management Module, and Reporting Module can be developed, as these modules rely on the database to store and retrieve data.
3. Product Management Module needs to be at least partially functional before Sales Management Module can be fully developed, as orders would require product information.
4. Purchase Management Module and Sales Management Module can be developed somewhat concurrently, but both should be operational before Reporting Module can be built, as reports need data from purchases and sales.
5. Security Implementation can be done in parallel with most other development tasks, but should be finalized before deployment.
6. User Interface Design & Development depends on the functionalities being developed in other modules, and ideally progresses iteratively alongside them.

**Critical Path**:

Based on the dependencies, the critical path (longest sequence of dependent tasks) likely consists of:

* Database Design
* Product Management Module (Core Functionalities)
* Purchase Management Module
* Sales Management Module
* Critical Reports (e.g., Stock Levels, Sales Reports)
* Security Implementation (integrated throughout)

The total critical path duration ranges from 12 to 17 weeks, suggesting this is the minimum time possible to complete the core functionalities of your inventory management system. However, remember that:

* This is an estimation and actual time may vary.
* Other tasks outside the critical path can still be worked on concurrently to optimize overall development time.
* Testing, bug fixing, and deployment will add additional time.

**5. Develop a schedule of all activities and tasks - weekly and monthly**.

Week 1-2 (Month 1):

* Activities:
  + System Requirements Gathering & Analysis
  + Database Design & Planning
* Tasks:
  + Conduct user interviews and surveys.
  + Define system functionalities and features.
  + Choose appropriate technology stack (framework, database).
  + Design initial database schema with tables and relationships.
  + Choose and install database software.

Week 3-4 (Month 1):

* Activities:
  + Product Management Module Development (Phase 1)
  + User Interface Design & Development (Phase 1)
* Tasks:
  + Develop core functionalities for adding, editing, and deleting products.
  + Design initial product management interface prototypes.
  + Start front-end development for product management forms and listings.

Week 5-6 (Month 1):

* Activities:
  + Product Management Module Development (Phase 2)
  + Security Implementation (Phase 1)
* Tasks:
  + Implement product image upload and management.
  + Design and implement basic user authentication and authorization.
  + Start securing data storage and communication channels.

Week 7-8 (Month 1):

* Activities:
  + Product Management Module Testing & Refinement
  + User Interface Design & Development (Phase 2)
* Tasks:
  + Conduct unit testing for product management functionalities.
  + Refine product management interface based on testing results.
  + Continue front-end development for product functionalities.

Week 9-10 (Month 2):

* Activities:
  + Purchase Management Module Development
  + Security Implementation (Phase 2)
* Tasks:
  + Develop functionalities for creating purchase orders.
  + Design and implement purchase order management interface.
  + Integrate with supplier APIs (if applicable).
  + Implement secure payment processing (if applicable).

Week 11-12 (Month 2):

* Activities:
  + Purchase Management Module Testing & Refinement
  + Reporting Module Development (Phase 1)
* Tasks:
  + Conduct unit testing for purchase management functionalities.
  + Refine purchase management interface based on testing results.
  + Design and develop basic inventory reports (stock levels, reorder points).

Week 13-14 (Month 2):

* Activities:
  + Sales Management Module Development (Phase 1)
  + Reporting Module Development (Phase 2)
* Tasks:
  + Develop functionalities for processing customer orders.
  + Design and implement sales management interface.
  + Integrate with payment gateway (if applicable).
  + Develop additional reports based on user requirements.

Week 15-16 (Month 2):

* Activities:
  + Sales Management Module Testing & Refinement
  + Reporting Module Testing & Refinement
* Tasks:
  + Conduct unit testing for sales management functionalities.
  + Refine sales management interface based on testing results.
  + Conduct unit testing for reports generation.
  + Refine reports based on testing results.

Week 17-18 (Month 3):

* Activities:
  + System Integration & Testing
  + User Acceptance Testing (UAT)
* Tasks:
  + Integrate all developed modules into a functional system.
  + Conduct comprehensive system testing for functionality and performance.
  + Conduct UAT with actual users to gather feedback and identify usability issues.

Week 19-20 (Month 3):

* Activities:
  + Bug Fixing & Refinement
  + Documentation & Training
* Tasks:
  + Address any bugs and issues identified during testing.
  + Prepare user manuals and training materials.
  + Conduct user training sessions.

Week 21-22 (Month 3):

* Activities:
  + Deployment & Monitoring
  + Post-Deployment Support
* Tasks:
  + Deploy the system to a production environment.
  + Monitor system performance and security.
  + Provide ongoing support to users and address any deployment issues.

Date

Signature of the student Signature of the cohort owner