**Department of Collegiate & Technical Education**

**Capstone Project**

**Format-2**

**Work Breakdown Structure**

**Capstone Project Name:** INVENTORY MANAGEMENT SYSTEM

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

DIVJA J [149CS21021]

DILEEP G P [149CS21020]

DHANYASHREE A P [149CS21019]

**Capstone Project Objective(s):** Implement a comprehensive Inventory Management System (IMS) to optimize stock levels, minimize costs, and improve operational efficiency, ultimately enhancing customer satisfaction and profitability.

* Implement real-time inventory tracking to ensure precise data on stock levels across all locations.
* Minimize discrepancies between physical and system inventory through regular cycle counts and reconciliations.
* Establish effective ordering strategies based on demand forecasting and lead times to prevent stockouts and overstocking.
* Implement reorder points and minimum/maximum stock levels to maintain optimal inventory balance.
* Automate manual inventory processes like receiving, picking, and shipping to enhance accuracy and speed.

**Work Breakdown Structure -Deliverables:**

**1. Planning and Design:**

1.1 Create Project Plan:

* Define project scope, goals, and objectives.
* Identify major project phases and milestones.
* Estimate project timeline and resources needed.
* Develop communication plan for stakeholders.
* Determine risk identification and mitigation strategies.
* Create initial budget estimation.

1.2 Gather and Document Requirements:

* Conduct stakeholder interviews and workshops.
* Analyze existing processes and workflows.
* Identify user needs and requirements.
* Prioritize requirements based on importance and complexity.
* Document requirements in a clear and concise format.

1.3 Design Database Schema:

* Define data entities and their relationships.
* Design tables with appropriate fields and data types.
* Establish primary and foreign key constraints.
* Normalize the schema to optimize performance.
* Document the schema design using diagrams and descriptions.

1.4 Create User Interface Mockups:

* Design wireframes or mockups for key screens and functionalities.
* Consider user flow and intuitive navigation.
* Incorporate branding and design guidelines.
* Obtain feedback from stakeholders on the mockups.

**2. Database Development:**

2.1 Create MySQL Database:

* Set up a MySQL database server or use a managed service.
* Create the database using appropriate naming conventions.
* Grant necessary permissions to users and applications.

2.2 Create Tables:

* Implement the designed tables based on the schema.
* Define data types and constraints for each table field.
* Set up primary and foreign key relationships.
* Consider performance optimization techniques like indexing.

2.3 Write Stored Procedures :

* Develop stored procedures for complex data manipulation tasks.
* Improve performance and security by encapsulating logic within the database.
* Test and optimize stored procedures thoroughly.

**3. Front-End Development:**

3.1 Create HTML Templates:

* Develop HTML templates for each page of the application.
* Use semantic markup and proper structure for accessibility.
* Incorporate reusable components and templates for efficiency.
* Integrate dynamic content using server-side scripting (PHP).

3.2 Write CSS Stylesheets:

* Define styles for text, colors, layout, and elements.
* Use responsive design principles for cross-device compatibility.
* Create a consistent and user-friendly visual experience.
* Consider using a CSS framework for faster development.

3.3 Write JavaScript Code :

* Implement interactive features and UI elements using JavaScript.
* Validate user input and handle form submissions.
* Use AJAX for dynamic data updates without page reloads.
* Ensure compatibility with different browsers and devices.

**4. Back-End Development:**

4.1 Write PHP Scripts for CRUD Operations:

* Develop PHP scripts for creating, reading, updating, and deleting data.
* Connect to the MySQL database using secure PDO methods.
* Validate and sanitize user input to prevent security vulnerabilities.
* implement business logic for inventory management operations.

4.2 Implement User Authentication:

* Develop user registration and login functionality.
* Store passwords securely using hashing and salting techniques.
* Implement session management for user identification.
* Assign different access levels and permissions based on user roles.

4.3 Develop Inventory Management Logic:

* Develop functionalities for adding, editing, and deleting inventory items.
* Implement logic for tracking stock levels, reordering points, and low stock alerts.
* Manage product categories and vendor relationships.
* Develop functionalities for managing purchase orders and invoices.

4.4 Create Reporting Functionality:

* Develop reports for inventory levels, sales trends, and purchase history.
* Allow users to filter and customize reports based on their needs.
* Export reports to different formats like PDF or CSV.

**5. Testing and Deployment:**

5.1 Write Test Cases:

* Create test cases for all functionalities of the system.
* Cover positive and negative test scenarios.
* Test user interface, database interactions, and business logic.

5.2 Conduct Testing:

* Manually test the system against the test cases.
* Utilize automated testing tools for regression testing.
* Fix any bugs and defects identified during testing.

5.3 Deploy System on a Web Server:

* Configure a web server environment (Apache, Nginx, etc.)
* Deploy the website files and scripts onto the server.
* Configure database connection settings and security.
* Test the deployed system in a production environment.

**6. Documentation:**

6.1 Write User Manual:

* Create a user manual for the inventory management system.
* Explain how to use the system's various features and functionalities.
* Include screenshots and step-by-step instructions for common tasks.

Date

Signature of the student Signature of the cohort owner