# Question 1 – Functional Requirements - 15 Marks

A functional requirement specifies what a system should do. functional requirement defines the specific behavior or features, or functions of a system and focuses on how the system interacts with users or other systems or data.

Functional requirement must be mapped to business requirements.

Example: The system should allow users to log in using their email and password.

A non-functional requirement defines the quality attributes or constraints of a system. non-functional requirement focuses on how the system performs a function rather than what the system does.

Example: The system should load the product catalog within 3 seconds under normal conditions.

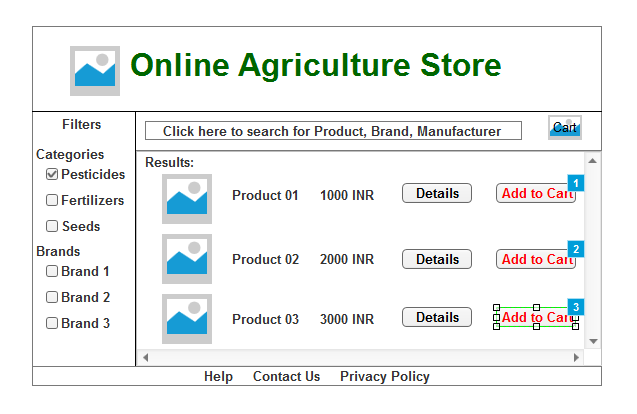
|  |  |  |  |
| --- | --- | --- | --- |
| **Req ID** | **Req Name** | **Req Description** | **Priority** |
| **FR001** | User Registration | New users (farmers and manufacturers) should be able to register by creating an account. | 10 |
| **FR002** | User Login | Existing users should be able to log in using their credentials. | 10 |
| **FR003** | Product Catalog | Farmers should be able to view a catalog of fertilizers, seeds, and pesticides. | 9 |
| **FR004** | Product Search | Farmers should be able to search for products by name, category, or manufacturer. | 9 |
| **FR005** | Product Upload | Manufacturers should be able to upload and manage product details like name, price, and description. | 8 |
| **FR006** | Add to Cart | Farmers should be able to add products to their cart for purchase. | 8 |
| **FR007** | Order Placement | Farmers should be able to place orders by selecting products from the cart. | 9 |
| **FR008** | Payment Gateway Integration | The application should allow farmers to pay via COD, UPI, or Credit/Debit card. | 9 |
| **FR009** | Order Tracking | Farmers should be able to track the status of their order (e.g., Processing, Shipped, Delivered). | 8 |
| **FR010** | Notifications | Users should receive notifications for order confirmation, shipment, and delivery status. | 7 |
| **NFR001** | System Uptime | The system should have an uptime of 99.9% to ensure availability to users in remote areas. | 10 |
| **NFR002** | Performance | The system should support at least 1,000 concurrent users without performance degradation. | 9 |
| **NFR003** | Security | User data and transactions must be encrypted and follow industry security standards. | 10 |
| **NFR004** | Scalability | The system should be scalable to handle increased traffic during peak agricultural seasons. | 9 |
| **NFR005** | Multi-Device Support | The application should be responsive and work seamlessly on both web and mobile devices. | 9 |
| **NFR006** | Load Time | The product catalog should load within 3 seconds under normal network conditions. | 8 |
| **NFR007** | Localization | The system should support multiple languages to cater to farmers from different regions. | 8 |
| **NFR008** | Data Backup | The system should back up data daily to prevent loss in case of server failure. | 8 |
| **NFR009** | Usability | The system interface should follow user-friendly design principles suitable for non-technical users. | 10 |
| **NFR010** | Compliance | The application should comply with relevant data protection and e-commerce regulations. | 9 |

# Question 2–Minimum 5 page designs - 15 Marks

## Login screen



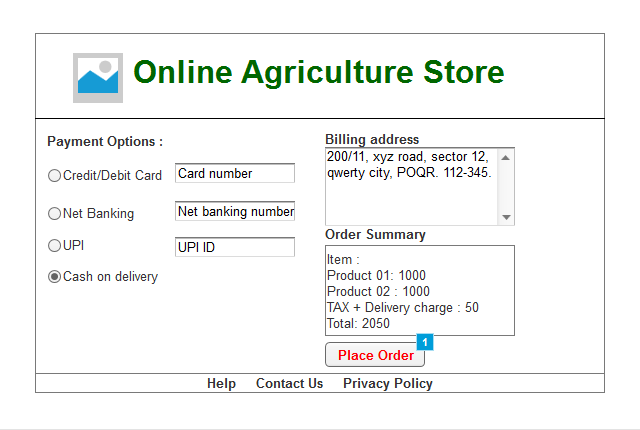
## Searching screen (search for the pesticides)



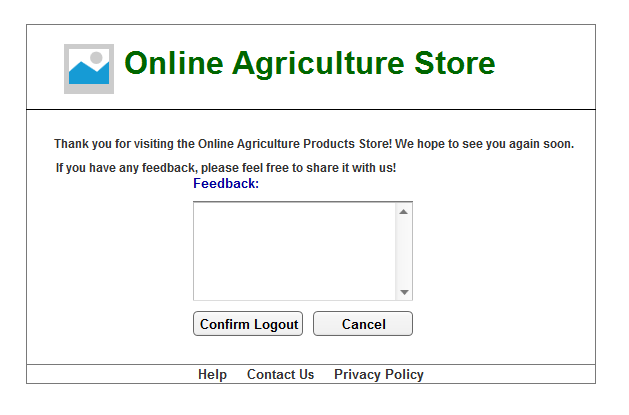
## Add to cart screen (add pesticides to the cart)



## Payment screen (make the payment for pesticides)



## Logout screen



# Question 3 – Tools (Visio, Balsamiq) - 15 Marks

## Visio

Microsoft Visio is a powerful diagramming tool used for creating flowcharts, organizational charts, network diagrams, and process maps. It supports high-quality visuals and detailed diagrams, making it ideal for technical professionals. Visio integrates seamlessly with Microsoft Office, offering advanced features for collaboration and customization

## Balsamiq

Balsamiq is an intuitive wireframing tool designed to create low-fidelity mockups of user interfaces. Its drag-and-drop simplicity and sketch-like style help teams visualize ideas quickly. Balsamiq fosters collaboration, enabling stakeholders to review and refine designs efficiently. It’s perfect for early-stage UI/UX planning and prototyping.

# Axure

Axure is a robust prototyping and wireframing tool used for designing interactive and functional user interface prototypes. It allows teams to create dynamic, high-fidelity mockups with advanced features like conditional logic, animations, and adaptive views. Axure is ideal for complex projects requiring detailed interactivity and user experience simulations.

# Question 4 – RTM - 6 Marks

Requirements Traceability Matrix is a complete list of requirements. Here we map functional requirement to business requirement. RTM has to update regularly with the daily data.



# Question 5 – 10 Test Case Documents - 10 Marks

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Test Case Name** | **Description** | **Test Steps** | **Expected Result** | **Status** |
| TC001 | Login Functionality | Verify that users can log in with valid credentials. | 1. Open login page. | User is logged in and redirected to the home page. | Passed |
|  |  |  | 2. Enter a valid username and password. |  |  |
|  |  |  | 3. Click "Login." |  |  |
| TC002 | Invalid Login Attempt | Verify that an error message is shown when entering incorrect credentials. | 1. Open login page. | Error message appears: "Invalid username or password." | Passed |
|  |  |  | 2. Enter invalid username/password. |  |  |
|  |  |  | 3. Click "Login." |  |  |
| TC003 | Search Pesticides | Verify that users can search for a pesticide using the search bar. | 1. Open search page. | Search results display all matching pesticides. | Passed |
|  |  |  | 2. Enter a pesticide name (e.g., "Neem Oil"). |  |  |
|  |  |  | 3. Click "Search." |  |  |
| TC004 | Filter Search Results | Verify that users can filter pesticide search results by category or price range. | 1. Open search page. | Filtered results display pesticides within the selected range. | Pending |
|  |  |  | 2. Apply filters (e.g., price range ₹100-₹500). |  |  |
|  |  |  | 3. Click "Apply Filter." |  |  |
| TC005 | Add to Cart | Verify that users can add pesticides to the shopping cart. | 1. Search for a pesticide. | The item is added to the cart and cart count updates. | Passed |
|  |  |  | 2. Click "Add to Cart" on the product. |  |  |
| TC006 | Update Cart Quantity | Verify that users can update product quantities in the shopping cart. | 1. Open cart page. | Product quantity updates and total cost recalculates accordingly. | Pending |
|  |  |  | 2. Change the quantity of a product. |  |  |
|  |  |  | 3. Click "Update." |  |  |
| TC007 | Payment Gateway | Verify that users can successfully complete a payment using a valid credit card. | 1. Add items to the cart. | Payment is processed successfully, and an order confirmation is displayed. | Pending |
|  |  |  | 2. Proceed to payment. |  |  |
|  |  |  | 3. Enter valid credit card details. |  |  |
|  |  |  | 4. Click "Pay Now." |  |  |
| TC008 | Payment Failure Handling | Verify that an appropriate error message is shown when payment fails (e.g., invalid card details). | 1. Add items to the cart. | Error message appears: "Payment failed. Please try again." | Pending |
|  |  |  | 2. Proceed to payment. |  |  |
|  |  |  | 3. Enter invalid credit card details. |  |  |
|  |  |  | 4. Click "Pay Now." |  |  |
| TC009 | Logout Functionality | Verify that users can log out successfully. | 1. Log in to the platform. | User is logged out and redirected to the login page. | Passed |
|  |  |  | 2. Click "Logout" in the menu. |  |  |
| TC010 | Session Expiry Handling | Verify that the user is logged out after a period of inactivity. | 1. Log in to the platform. | User session expires, and the login page appears with a message: "Your session has expired." | Pending |
|  |  |  | 2. Stay inactive for the defined session timeout duration. |  |  |

# Question 6 – DB Design – 8 Marks

### Database schema

Database schema is a combination of table and records, It is a show the relation between table records and entities.

|  |  |  |
| --- | --- | --- |
| **User Table** |  |  |
| **Column Name** | **Data Type** | **Description** |
| UserID | INT (Primary Key) | Unique identifier for each user. |
| Username | VARCHAR(50) | User's username. |
| Password | VARCHAR(100) | Hashed password. |
| Email | VARCHAR(100) | User's email address. |
| PhoneNumber | VARCHAR(15) | User's contact number. |
| Role | ENUM ('Farmer', 'Admin') | Role of the user. |
| CreatedAt | TIMESTAMP | Account creation date and time. |

|  |  |  |
| --- | --- | --- |
| **Products Table** |  |  |
| **Column Name** | **Data Type** | **Description** |
| ProductID | INT (Primary Key) | Unique identifier for each product. |
| Name | VARCHAR(100) | Name of the product. |
| Category | ENUM ('Pesticide', 'Fertilizer', 'Seed') | Product category. |
| Description | TEXT | Details about the product. |
| Price | DECIMAL(10,2) | Price per unit. |
| Stock | INT | Quantity available in stock. |
| CreatedAt | TIMESTAMP | Date and time the product was added. |

|  |  |  |
| --- | --- | --- |
| **Cart Table** |  |  |
| **Column Name** | **Data Type** | **Description** |
| CartID | INT (Primary Key) | Unique identifier for each cart. |
| UserID | INT (Foreign Key) | References the Users table. |
| ProductID | INT (Foreign Key) | References the Products table. |
| Quantity | INT | Quantity of the product added. |
| AddedAt | TIMESTAMP | Date and time the product was added. |

|  |  |  |
| --- | --- | --- |
| **Orders Table** |  |  |
| **Column Name** | **Data Type** | **Description** |
| OrderID | INT (Primary Key) | Unique identifier for each order. |
| UserID | INT (Foreign Key) | References the Users table. |
| TotalAmount | DECIMAL(10,2) | Total cost of the order. |
| OrderStatus | ENUM ('Pending', 'Completed', 'Cancelled') | Status of the order. |
| CreatedAt | TIMESTAMP | Date and time the order was placed. |

|  |  |  |
| --- | --- | --- |
| **OrderDetails Table** |  |  |
| **Column Name** | **Data Type** | **Description** |
| OrderDetailID | INT (Primary Key) | Unique identifier for the order detail. |
| OrderID | INT (Foreign Key) | References the Orders table. |
| ProductID | INT (Foreign Key) | References the Products table. |
| Quantity | INT | Quantity of the product ordered. |
| Price | DECIMAL(10,2) | Price of the product at the time of order. |

|  |  |  |
| --- | --- | --- |
| **Payments Table** |  |  |
| **Column Name** | **Data Type** | **Description** |
| PaymentID | INT (Primary Key) | Unique identifier for each payment. |
| OrderID | INT (Foreign Key) | References the Orders table. |
| PaymentMethod | ENUM ('Credit Card', 'Net Banking', 'UPI') | Payment type. |
| PaymentStatus | ENUM ('Success', 'Failed') | Payment status. |
| TransactionID | VARCHAR(100) | Unique identifier for the transaction. |
| PaidAt | TIMESTAMP | Date and time of payment. |

### Entity relationship

Entity relationship diagram is an image or figure which shows the relation between table and records is called as entity relationship diagram.

Users ↔ Orders: One user can place multiple orders (1:N).

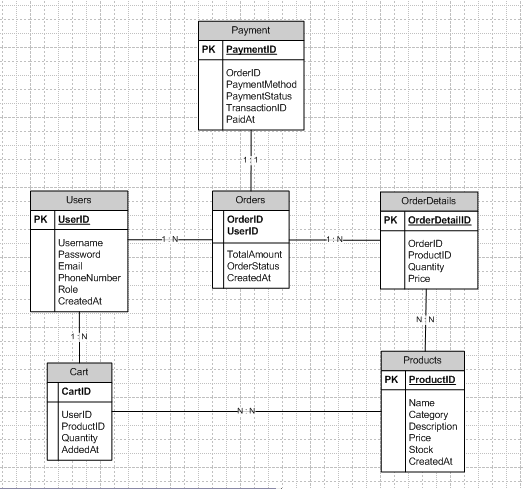
Orders ↔ OrderDetails: One order can have multiple products (1:N).

Products ↔ OrderDetails: A product can appear in multiple orders (N:N through OrderDetails).

Users ↔ Cart: One user can have one cart (1:N).

Cart ↔ Products: Multiple products can be added to a cart (N:N).

Order (1) → Payment (1): The OrderID column in the Payments Table acts as a foreign key referencing the OrderID from the Orders Table.



# Question 7 – Data Flow Diagram - 3 Marks

A data flow diagram (DFD) is a visual representation of how data moves through a system or process.

### External Entities:

**User (Farmer):** Interacts with the system to register, log in, search for items, add items to the cart, place orders, and make payments.

### Processes:

1. **Registration**: Allows new users to create an account with the platform.
2. **Login**: Authenticates users when they log in with their credentials.
3. **Account**: Users can manage their account details.
4. **Search Item**: Users search for products (e.g., pesticides, fertilizers) in the store.
5. **Buy Item**: Users add items to their shopping cart.
6. **Make Order**: Users place an order after adding items to the cart.
7. **Make Payment**: Processes payment for the placed order.

### Data Stores:

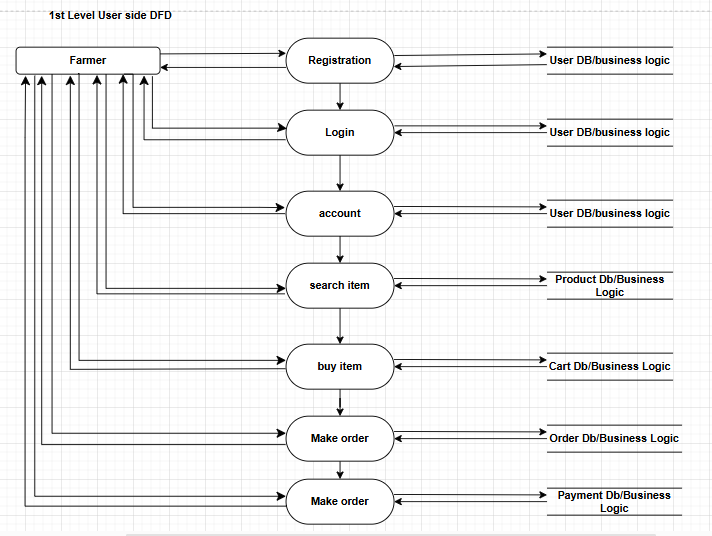
**User Data**: Stores user details (e.g., registration info, login credentials).

**Product Data**: Stores the product details (e.g., pesticides, fertilizers).

**Cart Data**: Temporarily stores the products added to the cart.

**Order Data**: Stores order details once the user places the order.

**Payment Data**: Stores payment transaction details once the payment is completed.



# Question 8 – Change Request - 10 Marks

CR are basically come under feasibly study. Handling CR is based on 3 parameter feasible study, Impacting analysis, Effort estimation

As a business analyst we should document the requirement and work with the development team to determine the feasibility and impact of the new feature. we should also consider the potential benefit, risk and cost associated with the enhancement before making any recommendations to the client.

1st we will understand the scope of change, What is an change ? how can we do accommodate new tax changes ? then we will identify it is a change request or new enhancement ? here it is change request as we have to modify the tax calculation logic which is already exist. Determine whether the change is inside/outside the scope. once the analysis is done will approve or reject CR and if it is approved then will communicate and implement the approved CR.

# Question 9 – Change Request Vs an Enhancement - 5 Marks

|  |  |
| --- | --- |
| **Change Request** | **Enhancement** |
| A formal request to modify existing functionality, processes, or requirements. | A request to add new features, improve performance, or extend the system's functionality. |
| To adapt the system to meet revised requirements or rectify existing issues. | To introduce improvements that enhance the system's usability, performance, or value. |
| Often arises due to external factors (e.g., regulatory changes, errors, or unforeseen needs). | Usually initiated internally or by stakeholders to add value or address user feedback. |
| Focuses on adjusting existing functionality within the current project or system scope. | Expands the project's or system's scope by adding new capabilities. |
| - Changing tax rules due to government regulations. | - Adding a recommendation engine to an online store. |
| - Updating login security protocols to align with compliance standards. | - Introducing a new reporting dashboard for better analytics. |
| Includes assessing how changes affect existing features, timeline, and budget. | Evaluates the cost, feasibility, and benefits of introducing new functionality. |
| Typically involves modifying or refining existing workflows, code, or designs. | May involve designing, developing, and integrating entirely new modules. |
| Often requires formal approval, especially if it impacts project scope, budget, or timeline. | Also requires formal approval but is often planned and prioritized during future releases. |
| Implemented as soon as possible, especially if it’s mandatory (e.g., compliance changes). | Planned for a future release based on resource availability and project roadmap. |
| Can cause budget overages if not accounted for in the original scope. | May require additional funding or approval for new resource allocation. |
| High priority, especially if related to legal, regulatory, or critical system issues. | Often medium to low priority unless it's a strategic improvement or critical to user satisfaction. |
| High risk if not implemented correctly, as it might disrupt existing functionality. | Lower risk but requires thorough testing to ensure compatibility with existing systems. |
| Includes a Change Request Form (CRF) with impact analysis and updated requirement documents. | Requires updated specifications, designs, and feature requests added to the backlog or roadmap. |

# Question 10 – Estimations - 6 Marks

In this case, they want the Farmers to sell their crop yields through this application i.e. Farmers should be able to add their crop yields and display to general public and should be able to sell them. They also want to introduce auction system for their crop yields this would be considered an enhancement as it involves adding new features to the system. As a Business Analyst, we should document the requirements and work with the development team to determine the feasibility and impact of the new features. we should also consider the potential benefits, risks, and costs associated with the enhancements before making any recommendations to the client.

### Man hours are the required effort of the resource to complete a project. There are three type of project

* Small: Upto 500 hours
* Medium: Upto 1000 hours
* Large: Upto 1500 hours

As per the case study, the duration of the project is 18 months and the current team size is around 15. This will come under medium project. As the trained resources are available, trainers are not required. As the structure of the project is available, New and enhanced infrastructure is not required.

### If the team works with the following allocation:

5 Developers: Focus on backend/frontend and database tasks.

2 UI/UX Designers: Work on interface design.

2 Testers: Handle unit, integration, and UAT testing.

1 Project Manager: Oversee progress and coordination.

### Assuming 8 hours/day of work:

Total team effort/day = 5+2+2+1 = 10 Resources × 8 = 80 hours/day

5+2+2+1=10 resources × 8 = 80hours/day

Time to complete = 1000/80 = 12.5

working days 1000/80=12.5working days

Thus, the enhancement can be completed within 2–3 weeks depending on priority and stakeholder approval.

# Question 11 – UAT – 6 Marks

User Acceptance Testing (UAT) is the final phase of the testing process in which the end-users or stakeholders validate the system or application to ensure it meets the business requirements and is ready for deployment. UAT focuses on verifying that the system behaves as expected in real-world scenarios and aligns with the original business objectives.

**Planning**: In this step, Blue Prints are made to implement UAT testing for every feature that needs to test and minimum standards for accepting the test.

**Designing**: Here the UAT Test cases are designed to hide all the possibilities of software packages in a real world environments.

**UAT Testers**: A Testing team consists of a end users that meet the criteria for implementing testing. The end user must have expertise in subject matter, the ability to report the problems.

**Bug Fixing**: Whatever Bugs are found in the UAT Testing, the development team should work on them and make it software error free.

**Sign Off:** After removing all the bugs, the testing team indicates acceptance of the completion of the bugs. In this phase, all the stakeholders come to a conclusion that the software is ready to GO LIVE and sign it off.

# Question 12 – Project Closure Document - 6 Marks

A Project Closure Document is a formal report that signifies the official completion of a project. It summarizes the project’s outcomes, evaluates its success against predefined objectives, identifies lessons learned, and secures formal sign-off from stakeholders. It serves as a final checkpoint to ensure all project deliverables are completed, approved, and accepted.

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No** | **Points to include** | **Details** | **Reference link** |
| 1 | **Did the client signed off on the UAT Testing** |  | Business Scope document |
|  | Date of the signed off: | 26th JAN 2025 |
|  | Name of the resource | Mr. Henery |
| 2 | **Objective of the project** |  |  |
|  | User friendliness | Achieved |  |
|  | Customer satisfaction | ROI in 6 months |  |
|  | More Categories | Achieved |  |
| 3 | **Functionalities worked on** |  |  |
|  | Secure payment processing | Achieved |  |
|  | Categories | Achieved |  |
| 4 | **Infrastructure** |  |  |
|  | Software installed | Achieved |  |
|  | Laptop purchased | Achieved |  |
| 5 | **Funding** |  | Finance Breakdown.docx |
|  | Amount approved | Rs. 2 Crore |
|  | Amount used | Rs. 1.5 Crore |
| 6 | **Overall Project information** |  |  |
|  | Escalations | 50 |  |
|  | Customer satisfaction | High |  |
| 7 | **Value to the company** |  |  |
|  | Positive/Negative | Positive - 90% -Company has gain an edge over the competitors -Increased Client -Trained employees -New project in pipeline |  |