**CAPSTONE PROJECT – PART 3**

1. **Identify 20 Functional Requirements**

Functional requirements define the specific behaviors,functions or operations of a system. They describe what the system should do, outlining the necessary tasks, actions or activities it must perform to achieve its objectives.

|  |  |  |
| --- | --- | --- |
| **Req ID** | **Req Name** | **Req Description** |
| FR001 | User Registration | Users should be able to register with the application. |
| FR002 | User Login | |  | | --- | |  |  |  | | --- | | Users should be able to log in with valid credentials. | |
| FR003 | Product Catalog | The application should display available fertilizers, seeds, and pesticides. |
| FR004 | Product Search | Farmers should be able to search for specific products using keywords or filters. |
| FR005 | Product Details Page | Farmers should be able to view detailed information about each product. |
| FR006 | Add to Cart | |  | | --- | |  |  |  | | --- | | Farmers should be able to add products to a shopping cart. | |
| FR007 | Checkout Process | |  | | --- | |  |  |  | | --- | | Farmers should be able to place an order by completing the checkout process. | |
| FR008 | Payment Integration | The application should support multiple payment methods (credit card, UPI, etc.) |
| FR009 | Order Tracking | Farmers should be able to track the status of their orders. |
| FR010 | Delivery Address Management | Farmers should be able to add, update, and save delivery addresses. |
| FR011 | Manufacturer Registration | Manufacturers should be able to register with the application. |
| FR012 | Manufacturer Login | Manufacturers should be able to log in with valid credentials. |
| FR013 | |  | | --- | |  |  |  | | --- | | Add Product | | Manufacturers should be able to upload product details (name, price, description). |
| FR014 | Update Product Details | Manufacturers should be able to update existing product information. |
| FR015 | View Orders | Manufacturers should be able to view orders placed for their products. |
| FR016 | Notifications for Orders | Manufacturers should receive notifications when orders are placed. |
| FR017 | Ratings and Reviews | Farmers should be able to rate and review purchased products. |
| FR018 | Customer Support | Farmers should be able to raise support tickets for any issues. |
| FR019 | Multilingual Support | The application should support multiple languages for better usability. |
| FR020 | Internet Connectivity Alerts | The app should alert users when they are offline and retry actions once connected. |

Non-functional requirements will describe the qualities and attributes of a system, focusing on how the system performs rather than specific behaviours or functions

|  |  |  |
| --- | --- | --- |
| **Req ID** | **Req Name** | **Description** |
| NFR001 | Usability | The application should have an intuitive and user-friendly interface, easily navigable by farmers with minimal training |
| NFR002 | Scalability | The system should handle a growing number of users and farmers |
| NFR003 | Performance | The application should load pages and respond to user actions within 2 seconds under normal load conditions. |
| NFR004 | Availability | The system should be available 99.9% of the time to ensure farmers and manufacturers can access it anytime. |
| NFR005 | Security | The application should ensure secure transactions and protect user data through encryption and secure authentication mechanisms. |
| NFR006 | |  | | --- | |  |  |  | | --- | | Accessibility | | The application should support multiple devices (mobile, tablet, desktop) and be compatible with low-speed internet connections. |
| NFR007 | Localization | The application should support multiple languages to cater to farmers in different regions of the country. |
| NFR008 | Maintainability | The system should be easy to update and maintain, with clear documentation for future enhancements. |
| NFR009 | Data Integrity | The system should ensure the accuracy and consistency of product and user data during all transactions. |
| NFR010 | Compliance | The application should comply with legal and regulatory requirements, including data protection and e-commerce regulations. |

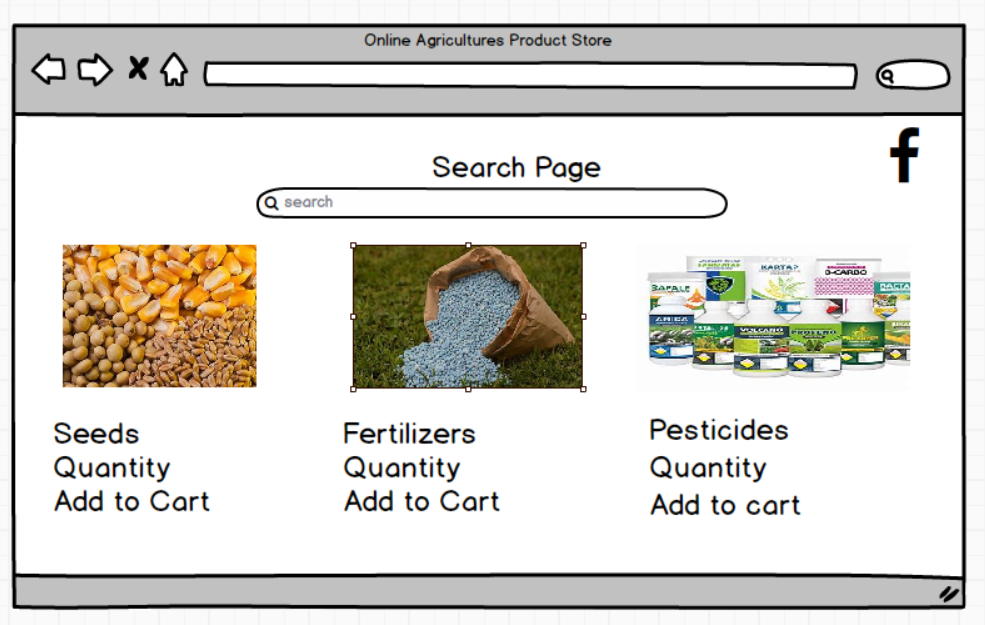
1. **Make wireframes and prototypes**
2. **Sign Up Page**



1. **Login Page**



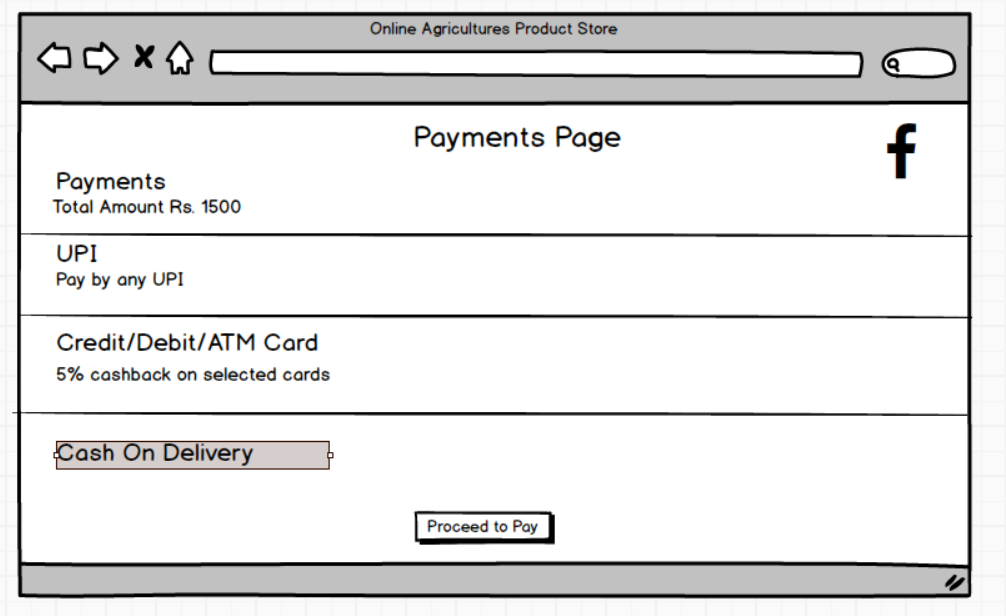
1. **Search Page**



1. **Cart Page**



1. **Payment Page**

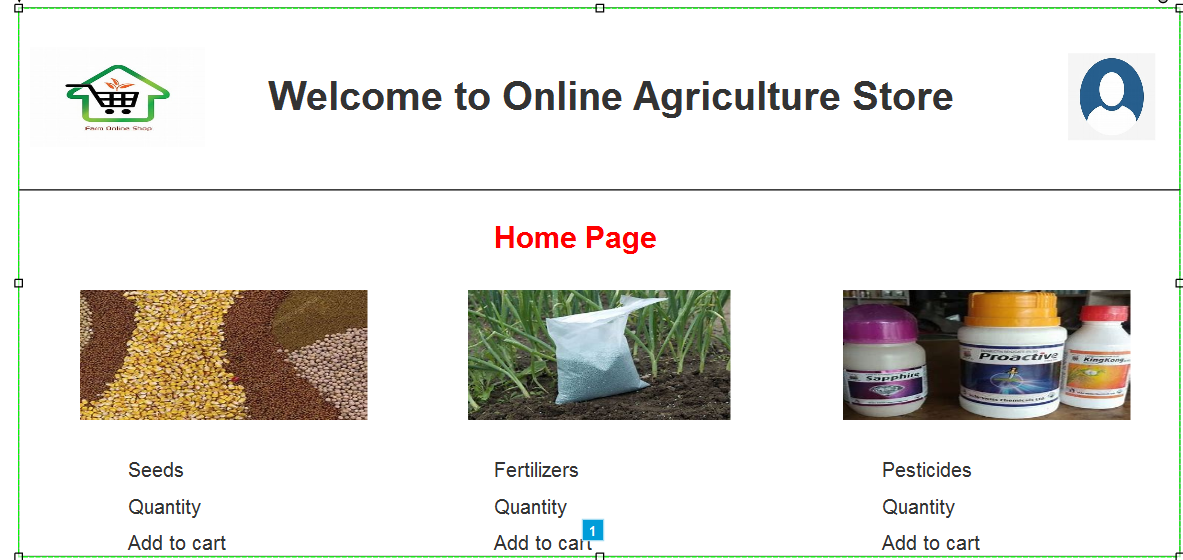


**PROTOTYPES**

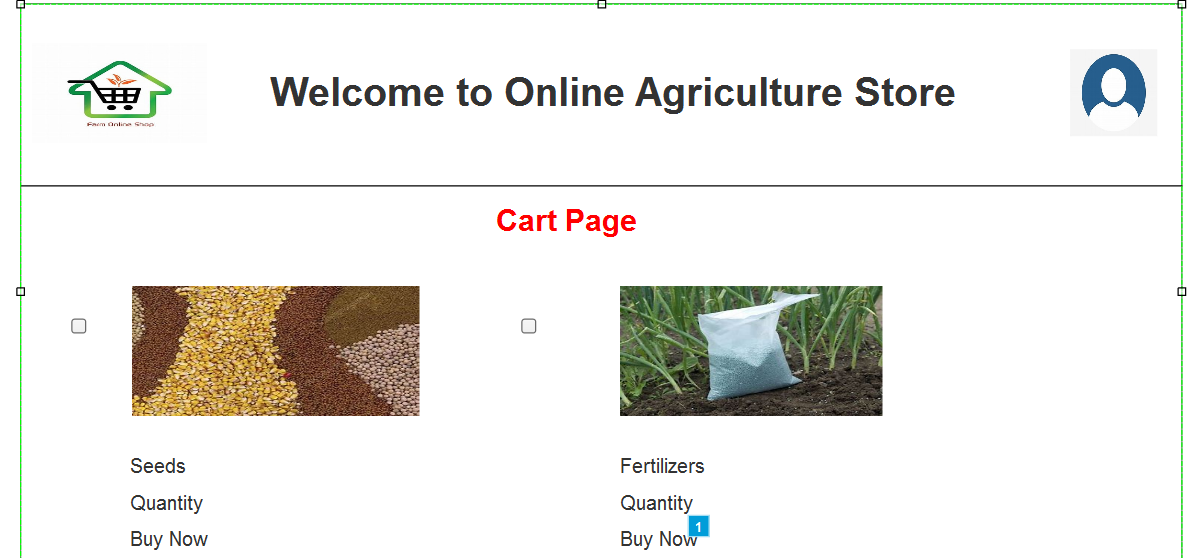
1. **Login Page**

****

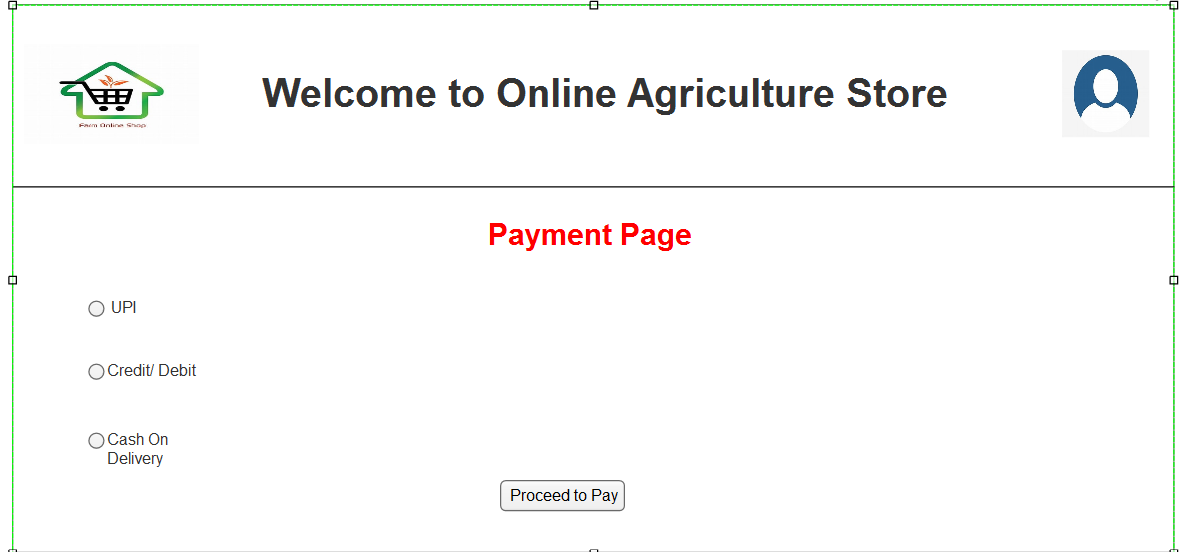
1. **Home Page**

****

1. **Cart Page**

****

1. **Payments Page**

****

**3. Make a note of the Tools, which you are using for above concepts.**

**MS Visio**

* Microsoft Visio is a diagramming and vector graphics application used to create professional diagrams, flowcharts, organizational charts, network diagrams, floor plans and more
* It features a drag-and-drop interface, making it easy to add shapes, connectors, and text to create visually appealing diagrams.
* Visio provides a variety of templates and stencils for different industries and use cases, including IT, business processes, engineering, and architecture.
* Visio allows multiple users to work on the same diagram simultaneously, improving team collaboration. It integrates with Microsoft Teams for enhanced communication.

**Balsamiq**

* Balsamiq is a rapid wireframing tool designed for creating low-fidelity mockups and prototypes of user interfaces for websites and applications.
* The tool features a drag-and-drop interface that mimics hand-drawn sketches, making it easy for both designers and non-designers to use.
* The tool intentionally limits visual design options (e.g., color schemes, advanced animations) to help teams focus on structure, layout, and user experience rather than aesthetics.
* Balsamiq provides a library of reusable UI elements, such as buttons, text fields, and menus, to quickly build prototypes

**Axure**

* Axure is a leading prototyping and wireframing tool used to create interactive and functional user interface prototypes.
* It allows users to build highly interactive prototypes with advanced functionality, including animations, conditional logic, and dynamic content.
* Axure enables creating complex interactions without requiring any coding skills, making it accessible to designers and product managers.
* Axure supports responsive design, allowing prototypes to adapt to different screen sizes, such as mobile, tablet, and desktop.

1. **Prepare RTM**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Req ID** | **Req Name** | **Req Description** | **Design** | **D1** | **T1** | **T2** | **T3** | **T4** | **UAT** |
| FR001 | Farmer Registration | Farmers should be able to register with the application |  |  |  |  |  |  |  |
| FR002 | Product Listing | Manufacturers should list products |  |  |  |  |  |  |  |
| FR003 | Product Browsing | Farmers should be able to browse listed products |  |  |  |  |  |  |  |
| FR004 | Order Placement | Farmers should be able to place orders |  |  |  |  |  |  |  |
| FR005 | Order Tracking | Farmers should be able to track orders |  |  |  |  |  |  |  |
| NFR0101 | Page loading time | Each page should load within 2 sec |  |  |  |  |  |  |  |
| NFR0102 | Security Compliance | Application should follow security practices |  |  |  |  |  |  |  |
| NFR0103 | Scalability | System should support atleast 10000 concurrent users |  |  |  |  |  |  |  |
| NFR0104 | Availability | Application should have 99.9% uptime |  |  |  |  |  |  |  |
| NR0105 | Mobile responsiveness | Application should work on mobile devices |  |  |  |  |  |  |  |

1. **Prepare 10 Test Case Documents**

Test Case 1: User Registration and Login

Test case ID: TC001

Description: Verify that farmers and manufacturers can register and log in successfully.

Pre-conditions: Internet connectivity, valid credentials.

Test Steps:

* Navigate to the application.
* Click on ‘Register’.
* Fill in user details (Name, Email, Phone, Address, User Type, etc.).
* Submit the registration form.
* Verify email confirmation.
* Log in with registered credentials.

Expected Result: User should be able to register and log in successfully.

Test Case 2: Manufacturer Adding Products

Test Case ID: TC002

Description: Verify that manufacturers can add products.

Pre-Conditions: Manufacturer must be logged in.

Test Steps:

* Navigate to the manufacturer dashboard.
* Click on ‘Add Product’.
* Enter product details (Name, Type, Price, Stock, Description, Image).
* Submit the form.
* Check if the product is listed.

Expected Result: Product should be visible in the store.

Test Case 3: Farmers Browsing Products

Test Case ID: TC003

Description: Verify that farmers can browse and filter products.

Pre-Conditions: At least one product should be available.

Test Steps:

* Navigate to the product catalog.
* Browse through the available products.
* Apply filters (Price, Type, Manufacturer, etc.)

Expected Result: Farmers should see relevant products based on filters.

Test Case 4: Adding Products to Cart

Test Case ID: TC004

Description: Verify that farmers can add products to the cart.

Pre-conditions: Farmer must be logged in, products must be available.

Test Steps:

* Navigate to a product page.
* Click ‘Add to Cart’.
* Verify product appears in the cart.

Expected Result: Product should be added to the cart successfully.

Test Case 5: Checkout and Payment

Test Case: TC005

Description: Verify that farmers can checkout and make payments.

Pre-Conditions: Farmer must have at least one product in the cart.

Test Steps:

* Navigate to the cart.
* Click ‘Proceed to Checkout’.
* Enter delivery details.
* Select payment method (UPI, Credit Card, COD, etc.).
* Confirm order

Expected Result: Order should be placed successfully.

Test Case 6: Order Tracking

Test Case: TC006

Description: Verify that farmers can track their orders.

Pre-Conditions: Order must be placed.

Test Steps:

* Navigate to ‘My Orders’
* Click on an order
* Check order status updates.

Expected Result: Farmers should see real-time order status.

Test Case 7: Product Reviews and Ratings

Test Case ID: TC007

Description: Verify that farmers can leave reviews and ratings.

Pre-Conditions: Farmer must have purchased a product.

Test Steps:

* Navigate to ‘My Orders’.
* Select a completed order.
* Click ‘Leave Review’.
* Enter rating and feedback.
* Submit review

Expected Result: Review should be visible on the product page.

Test Case 8: Admin Approving Products

Test Case ID: TC008

Description: Verify that an admin can approve or reject products before listing.

Pre-Conditions: Manufacturer must add a product.

Test Steps:

* Log in as admin.
* Navigate to pending products.
* Approve or reject a product

Expected Result: Only approved products should be visible to farmers.

Test Case 9: Security and Authentication

Test Case ID: TC009

Description: Verify that unauthorized users cannot access farmer or manufacturer accounts.

Pre-Conditions: Users must have registered accounts.

Test Steps:

* Attempt to log in with incorrect credentials.
* Attempt to access admin panel as a farmer.

Expected Result: Unauthorized access should be restricted.

Test Case 10: Mobile Responsiveness

Test Case ID: TC010

Description: Verify that the application works properly on mobile devices.

Pre-Conditions: Access to a mobile device.

Test Steps:

* Open the application on a mobile browser.
* Navigate through different sections.
* Perform major actions (Login, Add to Cart, Checkout, etc.).

Expected Result: Application should work seamlessly on mobile devices.

1. How should a BA handle change requests

As a Business Analysthandling Change Requestseffectively is crucial to ensure project success while maintaining scope, budget, and timeline.

1. **Establish a Clear Change Management Process**

Before handling change requests, a Change Control Process should be documented and agreed upon with stakeholders. This includes defining:

* Who can raise change requests?
* What information should be provided in a Change Request?
* How changes will be evaluated, approved, and prioritized?

1. **Capture and Document the Change Request**: Whenever a new requirementor change is requested by farmers, manufacturers, stakeholders, or theproject team, follow the below steps.

* Identify the requester: Who is requesting the change? (e.g., Peter, Kevin, Ben, Mr. Henry, SOONY Company, APT IT Solutions, etc.)
* Define the change: What is the nature of the request? (e.g., adding a new payment gateway, introducing a product recommendation system, integrating logistics tracking, etc.)
* Reason for change: Why is the change needed? (e.g., improving usability, regulatory compliance, business needs)
* Impact Analysis: How will it impact the existing requirements, system, cost,and timeline?

Example of Change Request template

|  |  |
| --- | --- |
| **Field** | **Description** |
| Requester | Who raised the change? |
| Change Summary | Brief description of the change |
| Reason for Change | Why is the change needed? |
| Impact Analysis | |  | | --- | |  |  |  | | --- | | Impact on scope, time, budget, resources | |
| Priority | |  | | --- | |  |  |  | | --- | | High/Medium/Low | |
| Approval Status | Pending/Approved/Rejected |

1. **Conduct Impact Analysis**

Before proceeding, analyze how the change affects:

* Project Scope - Does it require major modifications in functionalities?
* Timeline & Schedule - Will it extend the delivery deadline?
* Budget - Will it exceed the allocated **2 Crore INR**?
* Resources - Do we need additional developers, testers, infrastructure?
* Risk Assessment - Will it cause security, performance, or legal issues?

Example:

Change Request: Farmers request a feature to compare product prices across multiple manufacturers.

* Impact: Requires changes in UI/UX, additional API integrations, and database schema updates.
* Risk: Performance overhead due to price comparison across multiple vendors.
* Recommendation: Proceed if additional time & budget can be allocated.

1. **Seek Approval from the Project Sponsor:**

The Project Sponsor reviews the impact analysis and decides whether to

Approve - If the change aligns with business goals and is feasible.

Reject - If it adds too much complexity or cost.

Defer - If the change is beneficial but can be implemented later.

1. **Implement the Approved Change:**

If the change request is approved:

* Update the Business Requirement Document (BRD**)** & Functional Specification Document(FSD)
* Inform developers, testers, DB admin, and network team about the changes.
* Track the change implementation in JIRA, Confluence, or any PM tool.

1. **Validate and Test the Change**

* Testers (Mr. Jason & Ms. Alekya) will verify if the change meets the requirements.
* Conduct UAT (User Acceptance Testing) with Peter, Kevin, and Ben (Stakeholders).
* Deploy the change in staging environment before moving to production.

1. **Communicate and Document the Change**

* Update all relevant project documents.
* Inform farmers, manufacturers, and stakeholders about new features.
* Conduct training or documentation if required.