## Feature 1: Auth Feature

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Phase | Item No | Specification [Role] | User Story | Acceptance Criteria | Tags |
| Phase 1 | 1 | Register/Login [User] | - Users can register using name, email, and password. - Users can log in using valid credentials. | - Email must be unique - Password at least 8 characters - JWT generated on successful login - Error message if login fails | signup, signin |
| Phase 1 | 2 | Authentication [System] | - System stores hashed passwords securely. | - Passwords are hashed with bcrypt - Stored securely in the database | auth |

## Feature 2: Environment & API

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Phase | Item No | Specification [Role] | User Story | Acceptance Criteria | Tags |
| Phase 1 | 3 | API Responses [System] | - System provides consistent JSON responses for all endpoints. | - Success and error responses follow a standard structure - Helpful messages in case of failure | api |
| Phase 1 | 4 | .env Configuration [System] | - Developers manage environment secrets like database credentials and email configuration. | - All sensitive data stored in .env file - Not pushed to version control | config |

## Feature 3: Product Management

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Phase | Item No | Specification [Role] | User Story | Acceptance Criteria | Tags |
| Phase 2 | 5 | Product CRUD [Admin] | - Admins can create, read, update, and delete products. | - Fields: name, description, price, stock - Admin-only access - REST API implemented | product |
| Phase 2 | 6 | Product Filtering [User] | - Users can filter product listings by price. | - Query parameter filtering works - Invalid input handled properly | product-filter |

## Feature 4: Role-Based Access

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Phase | Item No | Specification [Role] | User Story | Acceptance Criteria | Tags |
| Phase 2 | 7 | Role-based Access [System] | - Admins manage the platform; users can only shop. | - Admin routes protected - JWT used to determine role - Unauthorized access returns 403 | access-control |

## Feature 5: Cart Management

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Phase | Item No | Specification [Role] | User Story | Acceptance Criteria | Tags |
| Phase 3 | 8 | Cart System [User] | - Users add/remove items from cart and view contents. | - Auth required - Cart tied to user ID - No duplicates, increase quantity if same product | cart |

## Feature 6: Order Processing

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Phase | Item No | Specification [Role] | User Story | Acceptance Criteria | Tags |
| Phase 3 | 9 | Order Placement [User] | - Users place orders from their cart and view history. | - Order saved with timestamp and items - Accessible in order history | order |

## Feature 7: Notification Email (Order Confirmation)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Phase | Item No | Specification [Role] | User Story | Acceptance Criteria | Tags |
| Phase 3 | 10 | Order Email [System] | - Users receive an email after placing an order. | - Email sent to user’s email using Nodemailer - Includes order details - Configured via .env | notification |