**Sprint Backlog for RevShop Project**

**Sprint 1: Project Setup and Basic User Management (Days 1-5)**

1. Project Initialization:

- Set up project repository and development environment.

- Configure basic project structure (folders for Java, HTML, CSS, and JS).

2. Frontend Setup:

- Create basic HTML/CSS layout for the home page and registration/login pages.

- Implement basic navigation and responsive design using Bootstrap.

3. Backend Setup:

- Set up a basic Java web application with Servlets and JSP.

- Create a JDBC connection to the SQL database.

4. \*\*User Registration and Login: \*\*

- Develop Servlets for user registration and login.

- Create JSP pages for registration and login forms.

- Implement basic input validation and error handling.

5. \*\*Database Setup: \*\*

- Design and create SQL tables for user management.

- Implement JDBC logic for storing and retrieving user data.

**Sprint 2: Product Management and Shopping Cart (Days 6-10)**

1. \*\*Product Management for Sellers: \*\*

- Develop Servlets and JSP pages for sellers to add, update, and delete products.

- Implement inventory management features.

2. \*\*Product Display for Buyers: \*\*

- Create HTML/CSS pages to display product details (image, price, description).

- Implement product browsing and search functionality.

3. \*\*Shopping Cart Functionality: \*\*

- Develop JavaScript for managing the shopping cart (add/remove products, update quantities).

- Create Servlets and JSP pages for the shopping cart and checkout process.

4. \*\*Database Updates: \*\*

- Design and implement SQL tables for product management.

- Update JDBC logic to handle product data.

5. \*\*Basic Testing: \*\*

- Perform basic testing of user registration, login, and product management features.

**Sprint 3: Order Processing and Notifications (Days 11-15)**

1. \*\*Order Processing: \*\*

- Develop Servlets for order placement and processing.

- Create JSP pages for checkout, order summary, and order history.

2. \*\*Notifications: \*\*

- Implement email notifications for order placement.

- Set up web notifications for sellers regarding low inventory.

3. \*\*Review and Favorited Features: \*\*

- Add functionality for buyers to review products and save them as favourites.

- Develop Servlets and JSP pages for product reviews and favourite management.

4. \*\*Security and Validation: \*\*

- Implement input validation and error handling for order and review features.

- Ensure basic security measures (e.g., input sanitization, HTTPS).

5. \*\*Performance Optimization: \*\*

- Optimize database queries and image assets for better performance.

**Sprint 4: Finalizing, Testing, and Deployment (Days 16-20)**

1. \*\*Final Touches: \*\*

- Refine UI/UX, fix bugs, and ensure consistency across pages.

- Perform final adjustments based on feedback and testing results.

2. \*\*Comprehensive Testing: \*\*

- Conduct end-to-end testing of all functionalities (registration, login, product management, cart, and checkout).

- Ensure unit tests and integration tests are written and passing.

3. \*\*Security and Performance: \*\*

- Implement additional security measures (e.g., SQL injection protection, secure credential storage).

- Validate the application using tools like Chrome’s Lighthouse.

4. \*\*Deployment Preparation: \*\*

- Prepare the application for deployment (e.g., set up cloud hosting, configure environment variables).

- Deploy a working version of the application on a cloud platform.

5. \*\*Technical Presentation and Documentation: \*\*

- Create a technical presentation to demonstrate the application’s features and architecture.

- Prepare associated diagrams and documentation for the project.

Sprint Summary\*\*

- \*\*Sprint 1:\*\* Focus on project setup and basic user management.

- \*\*Sprint 2:\*\* Implement product management and shopping cart functionality.

- \*\*Sprint 3:\*\* Develop order processing, notifications, and review features.

- \*\*Sprint 4:\*\* Finalize, test, and prepare for deployment.

--------------------------------------------------------------------------------------------------------------------------------------

### \*\*Sprint 1: Core Setup and Basic User Management (Days 1-5) \*\*

\*\*1. Project Initialization: \*\*

- \*\*Setup: \*\* Initialize version control (e.g., Git) and create project directory structure.

- \*\*Basic Configuration: \*\* Set up a basic Java web application with Maven/Gradle for dependency management.

\*\*2. Frontend Basics: \*\*

- \*\*HTML/CSS Layout: \*\* Create initial HTML structure and apply basic CSS styling.

- \*\*Bootstrap Integration: \*\* Incorporate Bootstrap for responsive design and layout.

\*\*3. Backend Basics: \*\*

- \*\*Servlets Setup: \*\* Create a basic Servlet to handle HTTP requests and responses.

- \*\*JSP Integration: \*\* Set up JSP pages for rendering HTML content dynamically.

\*\*4. Basic User Management: \*\*

- \*\*User Registration and Login: \*\* Implement basic Servlets for user registration and login.

- \*\*JSP Forms: \*\* Create simple registration and login forms.

- \*\*Validation: \*\* Implement basic client-side and server-side validation for user inputs.

\*\*5. Database Setup: \*\*

- \*\*SQL Schema: \*\* Design SQL tables for user management (e.g., users table).

- \*\*JDBC Connection: \*\* Implement JDBC for connecting to the database and performing CRUD operations for user data.

### \*\*Sprint 2: Intermediate Product Management and Cart (Days 6-10) \*\*

\*\*1. Product Management: \*\*

- \*\*Seller Features: \*\* Develop Servlets and JSP pages for sellers to add, update, and delete products.

- \*\*Product Forms: \*\* Create forms for product details (e.g., name, description, and price).

\*\*2. Product Display: \*\*

- \*\*Product Pages: \*\* Develop HTML/CSS pages to display products with details.

- \*\*Product Browsing: \*\* Implement basic product browsing by category or keywords using JavaScript.

\*\*3. Shopping Cart Basics: \*\*

- \*\*Cart Functionality: \*\* Develop JavaScript to manage cart operations (add/remove items).

- \*\*Cart Page: \*\* Create JSP pages to view and update cart contents.

\*\*4. Database Integration: \*\*

- \*\*Product Schema: \*\* Design and create SQL tables for product management.

- \*\*JDBC Updates: \*\* Extend JDBC logic to handle product-related CRUD operations.

\*\*5. Testing: \*\*

- \*\*Basic Unit Testing: \*\* Write unit tests for Servlets and Java components related to user management and product functionality.

### \*\*Sprint 3: Advanced Features - Orders, Reviews, and Notifications (Days 11-15) \*\*

\*\*1. Order Processing: \*\*

- \*\*Order Placement: \*\* Develop Servlets and JSP pages for the checkout process and order placement.

- \*\*Order Summary: \*\* Create order summary pages and order history views.

\*\*2. Notifications: \*\*

- \*\*Email Notifications: \*\* Implement email notifications for order confirmation.

- \*\*Web Notifications: \*\* Set up web notifications for low inventory (using JavaScript and backend logic).

\*\*3. Reviews and Favourites: \*\*

- \*\*Product Reviews: \*\* Develop functionality for buyers to review products.

- \*\*Favourites: \*\* Implement a system for saving products as favourites.

\*\*4. Security Enhancements: \*\*

- \*\*Input Validation: \*\* Implement robust server-side validation for order and review forms.

- \*\*Basic Security: \*\* Ensure secure handling of user inputs and sensitive data.

\*\*5. Performance Optimization: \*\*

- \*\*Optimize Queries: \*\* Review and optimize database queries for performance.

- \*\*Asset Compression: \*\* Compress images and other assets to improve page load times.

### \*\*Sprint 4: Finalization, Testing, and Deployment (Days 16-20) \*\*

\*\*1. Final Touches: \*\*

- \*\*UI/UX Refinements: \*\* Improve user interface and user experience based on feedback.

- \*\*Bug Fixes: \*\* Address any bugs or issues identified during previous sprints.

\*\*2. Comprehensive Testing: \*\*

- \*\*End-to-End Testing: \*\* Test all functionalities end-to-end, including registration, login, product management, cart, checkout, and notifications.

- \*\*Code Coverage: \*\* Ensure that test coverage is close to 80%.

\*\*3. Advanced Security: \*\*

- \*\*Injection Protection: \*\* Implement protection against SQL/NoSQL injection.

- \*\*Credential Storage: \*\* Ensure secrets and credentials are stored securely.

\*\*4. Deployment Preparation: \*\*

- \*\*Cloud Hosting: \*\* Prepare and configure cloud hosting for the application.

- \*\*Environment Variables: \*\* Set up environment variables and deployment configurations.

\*\*5. Presentation and Documentation: \*\*

- \*\*Technical Presentation: \*\* Prepare a presentation to demonstrate the application’s features and architecture.

- \*\*Documentation: \*\* Create documentation for the project, including system architecture, setup instructions, and user guides.

### \*\*Workflow Summary\*\*

- \*\*Sprint 1:\*\* Core setup and basic user management.

- \*\*Sprint 2:\*\* Intermediate product management and cart functionality.

- \*\*Sprint 3:\*\* Advanced features including order processing, reviews, and notifications.

- \*\*Sprint 4:\*\* Finalization, comprehensive testing, and deployment preparation.

This approach ensures that you build a strong foundation before moving on to more complex features, following a logical progression from basic to advanced concepts. If you need further adjustments or have any specific questions, feel free to ask!

Great! Let's dive into \*\*Sprint 1\*\* for the RevShop project. This sprint will focus on setting up the core project infrastructure and implementing basic user management functionalities. Here's a detailed breakdown of the tasks for Sprint 1, with estimated time allocations for each task to ensure you're on track:

### \*\*Sprint 1: Core Setup and Basic User Management (Days 1-5) \*\*

#### \*\*Day 1: Project Initialization\*\*

1. \*\*Project Repository and Environment Setup: \*\*

- \*\*Create Project Repository: \*\*

- Set up a Git repository for version control (GitHub, GitLab, and Bitbucket).

- \*\*Initialize Project: \*\*

- Create a new Java web project using your chosen build tool (Maven or Gradle).

- Set up directory structure (e.g., `/src/main/java`, `/src/main/webapp`, `/src/test/java`).

2. \*\*Basic Configuration: \*\*

- \*\*Configure Build Tool: \*\*

- Define dependencies for Servlets, JSP, JDBC, and any other necessary libraries in `pom.xml` (Maven) or `build.gradle` (Gradle).

- \*\*Set Up Development Environment: \*\*

- Configure IDE (e.g., IntelliJ IDEA, Eclipse) with necessary plugins and settings for Java web development.

#### \*\*Day 2: Frontend Basics\*\*

1. \*\*HTML/CSS Layout: \*\*

- \*\*Create Basic HTML Pages: \*\*

- Develop initial HTML pages for home, registration, and login.

- \*\*Apply Basic CSS Styling: \*\*

- Implement basic styling for layout and design consistency.

- \*\*Bootstrap Integration: \*\*

- Integrate Bootstrap framework to ensure responsive design.

2. \*\*Frontend Setup: \*\*

- \*\*Add Basic JavaScript: \*\*

- Include any necessary JavaScript libraries or frameworks (e.g., jQuery) if needed for future functionality.

#### \*\*Day 3: Backend Basics\*\*

1. \*\*Servlets Setup: \*\*

- \*\*Create Basic Servlet: \*\*

- Implement a simple Servlet to handle HTTP requests and responses (e.g., a welcome page).

- \*\*Configure Web.xml:\*\*

- Set up servlet mappings in `web.xml` or use annotations if applicable.

2. \*\*JSP Integration: \*\*

- \*\*Create JSP Pages: \*\*

- Develop basic JSP pages to dynamically render content (e.g., welcome.jsp).

- \*\*Integrate JSP with Servlets: \*\*

- Set up JSP pages to interact with Servlets for dynamic content generation.

#### \*\*Day 4: Basic User Management\*\*

1. \*\*User Registration and Login: \*\*

- \*\*Develop Registration Servlet: \*\*

- Create a Servlet to handle user registration, including form submission and validation.

- \*\*Create Registration JSP Page: \*\*

- Develop a JSP page with a registration form.

- \*\*Develop Login Servlet: \*\*

- Implement a Servlet to handle user login, including form submission and authentication.

- \*\*Create Login JSP Page: \*\*

- Develop a JSP page with a login form.

2. \*\*Validation and Error Handling: \*\*

- \*\*Implement Validation: \*\*

- Add basic client-side and server-side validation for user inputs (e.g., required fields, email format).

- \*\*Error Handling: \*\*

- Implement error messages for invalid input or login failures.

#### \*\*Day 5: Database Setup\*\*

1. \*\*Database Schema: \*\*

- \*\*Design SQL Tables: \*\*

- Create SQL tables for user management (e.g., `users` table with columns like `id`, `email`, `password`, `name`).

- \*\*Create Database: \*\*

- Set up the database schema in your SQL database.

2. \*\*JDBC Setup: \*\*

- \*\*Configure JDBC Connection: \*\*

- Implement JDBC connection logic to connect your Java application to the SQL database.

- \*\*CRUD Operations: \*\*

- Write basic CRUD operations for user data (e.g., insert new users, retrieve user details).

3. \*\*Testing Basic User Management: \*\*

- \*\*Test Registration and Login: \*\*

- Perform testing to ensure user registration and login functionalities are working correctly.

- \*\*Verify Database Integration: \*\*

- Confirm that user data is being stored and retrieved correctly from the database.

### \*\*Tasks Summary\*\*

- \*\*Day 1:\*\* Set up the project repository and development environment.

- \*\*Day 2:\*\* Develop basic HTML/CSS pages and integrate Bootstrap.

- \*\*Day 3:\*\* Implement basic Servlets and JSP integration.

- \*\*Day 4:\*\* Create user registration and login functionalities with validation.

- \*\*Day 5:\*\* Set up the database schema and implement JDBC for user data management.