MERN Stack Training
Company: Sensation Software Solutions
Student Name: Gurvinder Singh
Training Duration: 6 Months
Days: 84

Objective of the Day

The primary objective of **Day 84** was to design and implement a **Product Reviews system** for the GadgetShop e-commerce application. The aim was to allow authenticated users to share their feedback through star ratings and written comments, thereby improving customer trust, transparency, and engagement. This feature was planned to closely resemble real-world e-commerce platforms such as Amazon and Flipkart, where reviews play a critical role in influencing purchasing decisions.

Another important goal of the day was to integrate the reviews system smoothly into the existing **Product Details Page**, ensuring a seamless user experience without disrupting existing functionality such as product information display, add-to-cart flow, and authentication checks.

Introduction and Planning

At the beginning of the day, I analyzed the current structure of the GadgetShop project, particularly the **Product Details page**, API structure, and authentication flow. Since reviews are user-generated content, it was necessary to ensure that only **logged-in users** could submit reviews, while all users could view them.

I planned the reviews feature in two major parts:

1. **Backend logic** for storing, retrieving, and validating product reviews.
2. **Frontend UI and UX** for displaying existing reviews and submitting new ones.

Special attention was given to ensuring data integrity, preventing duplicate reviews by the same user, and maintaining performance while fetching reviews.

Backend Development – Reviews API

The backend development started with creating dedicated API routes for handling product reviews:

- **POST /products/:id/reviews** – To submit a new review.
- **GET /products/:id/reviews** – To fetch all reviews related to a specific product.

These routes were implemented following RESTful API standards and integrated into the existing Express.js routing structure.

Authentication and Authorization

To ensure secure access:

- Middleware was applied to verify **JWT authentication tokens**.
- Only authenticated users were allowed to post reviews.
- User identity (user ID and name) was extracted from the token and stored along with the review data.

This approach ensured accountability and prevented anonymous or spam reviews.

Review Data Structure

Each review included the following fields:

- User ID
- User Name
- Product ID
- Rating (1 to 5 stars)
- Review Comment
- Timestamp (createdAt)

This structure was designed to support future enhancements such as review editing, deletion, and moderation.

Validation and Error Handling

Backend validation was implemented to ensure:

- Rating value must be between **1 and 5**.
- Review comment cannot be empty.
- A user cannot submit multiple reviews for the same product.

Proper error messages were returned in case of validation failure, making it easier for the frontend to display user-friendly feedback.

Frontend Development – Reviews UI

Reviews Section Design

On the frontend, a dedicated **Reviews Section** was added to the Product Details page. This section was designed to appear below the product description and specifications, following common e-commerce UI patterns.

The reviews section included:

- Star rating display
- Reviewer name
- Review comment
- Review date

The layout was implemented using **React components** and styled using **Tailwind CSS**, ensuring consistency with the rest of the application.

Star Rating System

A reusable **Star Rating component** was created to visually represent ratings. This component:

- Displayed filled and unfilled stars based on rating value.
- Supported half-star logic for future scalability.
- Was used both for displaying existing reviews and selecting ratings during review submission.

Review Submission Form

A review submission form was displayed only to logged-in users. The form included:

- Star rating input
- Text area for review comment
- Submit button

Conditional rendering was used to hide the form from non-authenticated users, instead showing a message prompting them to log in.

Frontend Validation

Client-side validation was implemented to improve user experience:

- Rating selection was mandatory.
- Review comment could not be empty.
- Error messages were displayed instantly if validation failed.

This reduced unnecessary backend calls and ensured cleaner data submission.

Hands-on Practice and Testing

To validate the feature:

- Multiple reviews were submitted from different user accounts.
- Edge cases such as missing ratings and duplicate reviews were tested.
- Error handling responses were verified.

The reviews list was updated dynamically without requiring a page refresh, using React state updates after successful submission.

Average Rating Display

An important enhancement was calculating and displaying the **average rating** of each product:

- The average rating was shown on the Product Details page.
- It was also displayed on the **ProductCard component** used on the Shop and Home pages.

This provided quick visual feedback to users browsing multiple products.

UI Optimization and Responsiveness

The reviews UI was tested across different screen sizes:

- Desktop
- Tablet
- Mobile

Tailwind utility classes were used to ensure responsive spacing, readable text sizes, and proper alignment across devices.

Challenges Faced and Solutions

- **Duplicate Reviews:** Solved by checking user ID before allowing submission.
- **State Synchronization:** Solved using controlled components and state updates.
- **UX Consistency:** Solved by reusing components across pages.

Conclusion

Day 84 was a highly productive and feature-rich day in the GadgetShop project. The implementation of the **Product Reviews system** significantly enhanced user interaction and trust. By combining secure backend APIs with an intuitive frontend UI, the application now offers a real-world e-commerce experience. This feature added strong social proof to products and marked a major milestone in making the platform production-ready.

.