Retrospective

INSTRUCTIONS:

Answer the below questions. Remember to be honest about your experience. This will help the team make the necessary adjustments for the next sprint.

|  |  |
| --- | --- |
| Project Title: | Online Book Store System |
| Sprint #: | 1 |

|  |
| --- |
| What worked well this sprint? |
| * Sprint backlog was well-organized and helped guide the team. * Authentication UI and backend APIs were completed on time. * Smooth integration between frontend and backend for login flow. * Role-based access and OAuth worked as expected. |
| What didn’t work so well this sprint? |
| * Lack of clarity in choosing the right authentication method caused delays. * API structure was not finalized, leading to misalignment across teams. * Poor communication on session handling led to repeated debugging. * Testing setup and DB schema were inconsistent due to weak coordination. |
| What will we do to improve next sprint? |
| * Finalize key technical decisions early in planning. * Share detailed API structure before development begins. * Improve daily communication and documentation. * Start security and integration testing earlier. |

|  |  |
| --- | --- |
| Sprint #: | 2 |

|  |
| --- |
| What worked well this sprint? |
| * Book management API was completed and deployed. * CRUD and inventory logic worked efficiently. * Admin approval system was integrated. * UI bugs were tracked and resolved on time. |
| What didn’t work so well this sprint? |
| * Lack of clearly defined book categories led to confusion between design and backend teams. * UI designs were not finalized before development, causing rework. * Poor estimation of API performance impacted user experience. * Project manager oversight missed aligning frontend-backend milestones. |
| What will we do to improve next sprint? |
| * Lock design specifications before development starts. * Strengthen coordination between backend and UI teams. * Track milestone progress closely to avoid drift. * Optimize database and query performance. |

|  |  |
| --- | --- |
| Sprint #: | 3 |

|  |
| --- |
| What worked well this sprint? |
| * Cart and checkout flows were completed. * Payments and order status tracking worked end to end. * Admin monitoring of transactions was integrated. * Payment bugs were identified and resolved. |
| What didn’t work so well this sprint? |
| * Ambiguous cart update logic led to customer flow errors. * No clear strategy for mobile responsiveness. * Delayed planning for payment security and refund policies. * Lack of predefined transaction error handling caused bottlenecks. |
| What will we do to improve next sprint? |
| * Conduct payment sandbox testing in advance. * Improve mobile-first design planning. * Define transaction rules and refund scenarios upfront. * Collaborate earlier on risk scenarios. |

|  |  |
| --- | --- |
| Sprint #: | 4 |

|  |
| --- |
| What worked well this sprint? |
| * Subscription APIs and interfaces were developed. * Seller earnings dashboard showed correct metrics. * Bug fixes improved access logic and UI flow. |
| What didn’t work so well this sprint? |
| * Pricing strategy across seller plans was unclear from a business perspective. * Upgrade/downgrade workflows were not mapped before dev began. * Billing logic created confusion due to late involvement of stakeholders. * Inconsistent updates in earnings reports led to delays in QA. |
| What will we do to improve next sprint? |
| * Align pricing strategy and dev logic early with stakeholders. * Define subscription lifecycle rules before coding. * Improve coordination between finance logic and reporting. |

|  |  |
| --- | --- |
| Sprint #: | 5 |

|  |
| --- |
| What worked well this sprint? |
| * Backend and frontend modules for discounts were functional. * Price comparison with external sources worked with 90% accuracy. * Admin and seller discount roles were defined. |
| What didn’t work so well this sprint? |
| * External API rate limits were not planned for in initial sprint backlog. * No clear UX plan for how to display price comparisons. * Discount rules were overlapping due to lack of validation logic. * Promotions conflicted because of no approval workflow for sellers. |
| What will we do to improve next sprint? |
| * Plan API dependencies in backlog grooming. * Introduce seller-side validation for discounts. * Cache data to handle API limits better. |

|  |  |
| --- | --- |
| Sprint #: | 6 |

|  |
| --- |
| What worked well this sprint? |
| * Admin dashboard layout was finalized. * Seller approvals and access workflows were integrated. * Subscription tracking and analytics were built. |
| What didn’t work so well this sprint? |
| * Admin permissions were not structured, leading to over-access issues. * Unclear flow of seller approval caused confusion in the team. * Analytics chart logic and performance were not reviewed before dev started. * Subscription statuses in admin panel were not in sync due to misaligned APIs. |
| What will we do to improve next sprint? |
| * Refactor role hierarchy early and get it reviewed. * Define data sync and dashboard views before development. * Avoid mid-sprint changes in reporting features. |

|  |  |
| --- | --- |
| Sprint #: | 7 |

|  |
| --- |
| What worked well this sprint? |
| * Verified reviews and rating features were integrated. * Review API and filtering worked reliably. * Pagination logic was added later to stabilize long lists. |
| What didn’t work so well this sprint? |
| * UI glitches were not caught earlier due to incomplete test cases. * Review ratings and timestamps lacked QA review. * Pagination logic was not finalized before dev, causing missing reviews. * Communication gap between QA and frontend delayed bug fixes. |
| What will we do to improve next sprint? |
| * Prioritize review features in QA checklist. * Improve sync between dev and testing teams. * Finalize backend logic early to avoid pagination errors. |

|  |  |
| --- | --- |
| Sprint #: | 8 |

|  |
| --- |
| What worked well this sprint? |
| * Load testing scripts and optimization were completed. * Final patching and UI refinements improved user experience. * Feature testing before production was successful. |
| What didn’t work so well this sprint? |
| * Performance testing started too late in the cycle. * Dark mode and mobile view had design inconsistencies. * Security issues in session handling weren't caught early. * Deployment plan missed pre-launch staging verification. |
| What will we do to improve next sprint? |
| * Perform end-to-end testing in staging before launch. * Include mobile and accessibility in final QA scope. * Freeze codebase earlier to avoid last-minute bugs. |