

Checklist

Online ticket booking application

1. Searching

[nazia basbwjshw](#)

- Functional Testing : ensures that search filters and inputs return correct and expected results.
- Performance Testing: measures how fast search results load under different conditions.
- Usability Testing: verifies that users can easily use the search feature.
- Compatibility Testing: checks that search works properly across different browsers and devices.
- Black-Box Testing: tests search from the user perspective without knowing internal code.
- White-Box Testing: inspects internal search algorithms and logic.
- Grey-Box Testing: combines user perspective and partial code knowledge to test edge cases.
- Unit Testing: tests individual search functions like fetching information from database.
- Integration Testing: verifies how search integrates with getting data and filters
- System Testing: tests the entire search feature.

2.Booking Process

- Functional Testing: ensures booking steps like selecting seats and confirming tickets work correctly.
- Security Testing: verifies users cannot book using unauthorized access.
- Usability Testing: ensures users can complete bookings easily without confusion.
- Black-Box Testing: tests the booking process from the user interface without viewing internal code.
- White-Box Testing: inspects booking logic.
- Unit Testing: tests individual booking-related functions like how to reserve seat.
- Integration Testing: checks how booking works together with search and payment modules.
- System Testing: tests the full booking workflow.

3.Payment Process

- Functional Testing: confirms payments are processed correctly and users get confirmation.
- Security Testing: ensures payment details are protected and transmitted securely.
- Integration Testing: checks proper communication with payment.
- Performance Testing: measures how fast and reliably payments go through under load.
- Regression Testing: verifies that new changes do not break existing payment functions.
- White-Box Testing: examines internal code for calculations, encryption, and logic.
- Black-Box Testing: tests payments by simulating user behavior.
- Unit Testing: tests small parts like to validate card number or process payment.
- Integration Testing: checks payment works well with booking and refund modules.
- System Testing: tests full payment flow in the live-like environment.

4. Notifications

- Functional Testing: ensures notifications are sent correctly after key actions.
- Usability Testing: ensures the message content is clear, timely, and user-friendly.
- Load Testing: checks how well the system handles sending many notifications at once.
- White-Box Testing: inspects how messages are triggered and sent from the backend.
- Black-Box Testing: checks that the user receives correct notifications without seeing internal logic.
- Integration Testing: verifies communication with SMS, email, or push notification services.
- System Testing: tests notification delivery across the entire application.

5. User Account Management

5.1. Login

5.2. Sign-Up

5.3. Profile Updates

- Functional Testing: confirms all user actions like registration, login, and profile edits work.
- Security Testing: protects user data, passwords, and restricts access to personal information.
- Usability Testing: ensures account actions are easy to perform and understand.
- White-Box Testing: examines logic for password handling, authentication, and sessions.
- Black-Box Testing: tests user flows without knowing internal code.
- Unit Testing: tests components like validate password, create user.
- Integration Testing: checks that login, user profile, and session data interact correctly.
- System Testing: tests the complete account management process.

6. Cancelling and Refunds

- Functional Testing: ensures users can cancel tickets and get valid refund responses.
- Security Testing: verifies only authorized users can cancel their own bookings.
- Usability Testing: checks the process is easy to follow and messages are clear.
- White-Box Testing: examines refund calculations, status changes, and rules.
- Black-Box Testing: tests the cancellation process from the user's point of view.
- Unit Testing: tests functions like cancel booking and calculate refund.
- Integration Testing: ensures cancellation updates tickets, payments, and notifications correctly.
- System Testing: validates the full cancel and refund workflow.

7. Ticket History

- Functional Testing: verifies users can see their previous and upcoming bookings.
- Usability Testing: ensures the ticket history is easy to read and organized.
- Security Testing: prevents unauthorized access to others' ticket records.
- White-Box Testing: checks how ticket history is retrieved from the backend.
- Black-Box Testing: tests if correct ticket data is shown without knowing internal logic.
- Unit Testing: tests methods like to get user history.

- Integration Testing: verifies data consistency between bookings and user accounts.
- System Testing: validates the entire ticket history view for correctness and access control.

8.Review and Ratings

- Functional Testing: ensures users can write, update, and view reviews properly.
- Usability Testing: verifies the review/rating is clear and easy to use.
- Security Testing: prevents spam, abuse, and restricts reviews to real customers.
- White-Box Testing: checks how reviews are validated, stored, and linked to bookings.
- Black-Box Testing: simulates writing and reading reviews from the frontend.
- Unit Testing: tests individual components like save review, validate rating.
- Integration Testing: ensures reviews link correctly with user and trip/event data.
- System Testing: validates the complete review system.