Topic : Event Booking System

Phase 2: Development of the Minimum Viable Product (MVP)

Group Names :

Hitkumar Patel : Team Lead

Hrishikesh More : Backend Developer

Rudra Patel : Frontend Developer

Jagruti Kataria : Tester

1. Business Case Document

Problem Statement

Event organizers often face difficulties in managing ticket sales manually, leading to overbooking, lack of proper records, and security issues. Attendees also struggle to find and book tickets efficiently. A centralized event ticket booking system is required to streamline event listings, ticket sales, and attendance tracking.

Target Audience

- **Event Organizers**: Need a platform to create and manage event listings and track ticket sales.
- **Users/Attendees**: Need a seamless way to browse and purchase event tickets.
- **Admins**: Manage users, events, and transactions.

Proposed Solution

A Laravel-based Online Event Ticket Booking System that allows users to browse events, book tickets, and receive digital confirmations via email. Organizers can create and manage events, while admins oversee the system.

Scope

Core Features (MVP)

- ✅ **User Authentication** (Admin, Organizer, Attendee)
- ✅ **Event Management** (Create, Edit, Delete events)
- ✅ **Ticket Booking System** (Select event, book ticket, confirm booking)
- ✅ **Payment Integration** (Stripe/PayPal for ticket purchases)
- ✅ **Email Notifications** (Booking confirmation, event reminders)
- ✅ **Booking Updates** (Modify ticket quantities with additional payment flow)
- ✅ **Booking Cancellation** (Cancel bookings with refund records)
- ✅ **Activity Logging** (Track user actions and system events)
- ✅ **Automated Testing** (Unit and Feature tests for core functionality)

Optional Enhancements (For Future Implementation)

- 🚀 Social Media Sharing (Share event details on Facebook, Twitter)
- 🚀 Reviews & Ratings (Users can rate and review events)
- 🚀 Discount Coupons & Promo Codes (For promotional events)
- 🚀 QR Codes for Ticket Verification

2. Implementation Overview

Implemented User Roles

The system currently supports three distinct user roles:

1. **Admin**
   - Manage all users (view, edit roles)
   - Oversee all events (approve/reject, view details)
   - Access reporting dashboard with system-wide statistics
   - Monitor booking and payment activities

2. **Organizer**
   - Create and manage events
   - Create various ticket types with different prices
   - View bookings for their events
   - Generate event-specific reports
   - Publish/unpublish events

3. **Attendee**
   - Browse and search for events
   - View event details and available ticket types
   - Make bookings with secure payment
   - View, update, and cancel existing bookings
   - Receive email notifications for booking activities

Core Functionality Implemented

Event Management

- Event creation with detailed information (title, description, date, time, location)
- Image upload for event thumbnails
- Event publication control (draft/published states)
- Multiple ticket types per event with different pricing
- Inventory management for available tickets

Booking System

- Secure ticket reservation process
- Inventory verification to prevent overselling
- Booking modification with proper inventory adjustment
- Ticket quantity updates with additional payment handling
- Cancellation with automatic ticket inventory restoration
- Email confirmations for all booking activities

Payment Processing

- Integrated Stripe for credit card payments
- PayPal integration for alternative payment option
- Transaction recording and payment verification
- Additional payment flow for booking updates
- Refund recording for booking cancellations or quantity reductions

Security & Logging

- Secure authentication and role-based access control
- Comprehensive event logging for:
    - Authentication events (login, logout, failed attempts)
    - CRUD operations (create, update, delete actions)
    - Payment activities
    - System errors and exceptions
- Separate log channels for different event types
- Session management for payment processes

Data Integrity

- Database transactions to ensure data consistency
- Race condition prevention for concurrent ticket booking
- Validation of all user inputs
- Proper error handling with user-friendly messages

3. System Architecture

The system follows the MVC (Model-View-Controller) architecture pattern with:

Models

- User (with Admin, Organizer, Attendee subclasses)
- Event
- TicketType
- Booking
- Payment
- Notification

Controllers

- AuthController
- EventController
- BookingController
- PaymentController
- AdminController
- OrganizerController

Services

- LoggingService (for centralized logging)

Middleware

- AuthLogging (for authentication event tracking)
- Role-based access control

4. Technical Implementation

Backend

- Laravel 11.x framework
- PHP 8.x
- MySQL database with Eloquent ORM
- RESTful routing architecture
- Blade templating engine

Security

- Authentication with Laravel's built-in mechanisms
- CSRF protection for all forms
- Input validation and sanitization
- Database query optimization to prevent SQL injection

Payment Gateway Integration

- Stripe API for credit card processing
- PayPal API for alternative payments
- Webhook handling for payment notifications

Logging and Monitoring

- Custom logging service for all system events
- Multiple log channels for different event types
- Environment-specific logging configuration

5. Testing and Quality Assurance

Automated Testing Implementation

The system includes comprehensive automated tests to ensure functionality, reliability, and maintainability:

Unit Tests

- **Model Tests**: Verify model relationships and attributes
  - BookingTest: Tests booking relationships with users, events, ticket types
  - EventTest: Tests event relationships, date handling, and status flags

Feature Tests

- **Authentication Tests**:
  - Login screen rendering
  - User authentication with valid credentials
  - Authentication failure with invalid credentials
  - Registration functionality

- **Booking Flow Tests**:
  - Viewing booking pages
  - Creating new bookings
  - Viewing booking details
  - Cancelling bookings
  - Updating booking quantities with appropriate payment handling

- **Event Management Tests**:
  - Creating events
  - Editing events
  - Deleting events
  - Publishing/unpublishing events
  - Access control (preventing unauthorized edits)

- **Payment Tests**:
    - Payment page rendering
    - Payment processing
    - Notification creation
    - Access control for payment pages

Test Results

All 31 tests are passing successfully with 62 assertions, verifying the system's functionality across all core components:

- 9 Unit tests confirming model integrity
- 22 Feature tests validating end-to-end workflows

Test-Driven Development

The development process incorporated test-driven methodologies:

- Tests were written to define expected behavior
- Issues were identified through failed tests
- Fixes were implemented and verified through passing tests
- Additional tests were added to prevent regression

6. Business Justification

Value Proposition

- **For Organizers**: Simplified event management and ticket sales with reduced administrative overhead
- **For Attendees**: Convenient ticket purchase and management experience
- **For Platform**: Potential revenue through transaction fees or subscription model

Expected Benefits

- Reduced manual effort in event management
- Eliminated double-booking and inventory issues
- Enhanced user experience through digital ticket management
- Improved reporting and analytics for decision-making
- Secure payment processing with proper financial records

Risk Assessment

- **Competition**: Many existing event management platforms
- **Technical Risks**: Payment gateway dependencies
- **Mitigation**: Focus on ease of use and specific features for target audience

7. Future Development Roadmap

Short-term Enhancements

- QR code implementation for ticket verification
- Mobile responsive design optimizations
- Performance improvements for high-volume events

Medium-term Features

- Social media integration for event sharing
- User ratings and reviews for events
- Promotional code and discount functionality

Long-term Vision

- Mobile application development
- Advanced analytics for organizers
- Integration with external calendar systems
- Multi-language support

8. Conclusion

The Event Booking System has been successfully implemented with all core MVP features, including the additional payment flow for booking updates. The system provides a comprehensive solution for event management, ticket sales, and attendee management. It meets the requirements outlined in the project specification document and is ready for deployment.

The implementation includes extensive automated testing with 31 tests covering both unit and feature functionality, ensuring the system's reliability and maintainability. The test suite validates core system components and user workflows, providing confidence in the system's functionality and helping prevent regression issues during future development.