Sri Lanka Institute of Information Technology

Information Technology Project (IT2080)

Batch ID - Y2.S2.WD.IT.04
Group ID - 92

Pearlora – Travel Agency Management System

Activity 03

	Registration Number	Name with Initials	Email	Contact Number
1.	IT23273580	Prasanth A	it23273580@my.sliit.lk	0779080621
2.	IT23210424	Thivyan J	it23210424@my.sliit.lk	0711050671
3.	IT23210974	Kesigan M	it23210974@my.sliit.lk	0740422764
4.	IT23321236	Nafy F A	it23321236@my.sliit.lk	0722203811
5.	IT23158436	Priyatharshini A H	it23158436@my.sliit.lk	0702455393

Destination Management (IT23273580 – Prashanth A)

Epic 1: Destination Management System

Feature 1: Destination Search & Selection

User Story 1.1:

As a traveler, I want to browse available destinations so that I can choose the best place to visit.

- Task 1.1.1: Design and develop the Destination Selection UI.
- Task 1.1.2: Implement a search and filter API to fetch destinations based on category, location, and rating.
- Task 1.1.3: Connect the UI with the API to display available destinations.

User Story 1.2:

As a traveler, I want to view detailed destination information so that I can plan my trip easily.

- Task 1.2.1: Create a destination detail page with images, attractions, maps, and nearby facilities.
- Task 1.2.2: Integrate weather updates and real-time travel alerts.
- Task 1.2.3: Display reviews and ratings from previous visitors.

Feature 2: Smart Destination Recommendations

User Story 1.3:

As a traveler, I want to get AI-based recommendations so that I can find destinations that match my interests.

- Task 1.3.1: Develop an AI-powered recommendation engine based on past searches, preferences, and trends.
- Task 1.3.2: Implement personalized suggestions on the homepage.
- Task 1.3.3: Show top-rated and trending destinations dynamically.

User Story 1.4:

As a traveler, I want to compare multiple destinations so that I can make the best decision.

- Task 1.4.1: Create a destination comparison tool with side-by-side details.
- Task 1.4.2: Implement a "Save to Favorites" option.
- Task 1.4.3: Allow users to share destination lists with friends.

Epic 2: Traveler Engagement & Experience

Feature 3: Reviews & Ratings

User Story 2.1:

As a traveler, I want to leave reviews and rate destinations so that others can learn from my experience.

- Task 2.1.1: Create a review submission form.
- Task 2.1.2: Implement a rating system (stars or points).
- Task 2.1.3: Allow travelers to upload photos and videos in reviews.

User Story 2.2:

As a traveler, I want to read reviews from others so that I can choose the best destination.

- Task 2.2.1: Develop a review section with filters for most helpful, latest, and highest-rated reviews.
- Task 2.2.2: Enable commenting and upvoting on reviews.
- Task 2.2.3: Implement a "Report Review" feature for moderation.

Feature 4: Real-Time Tracking & Travel Assistance

User Story 2.3:

As a traveler, I want real-time location tracking so that I can navigate safely.

- Task 2.3.1: Integrate Google Maps API for live tracking.
- Task 2.3.2: Develop a "Find Nearby Attractions" feature.
- Task 2.3.3: Enable location sharing with family and friends.

User Story 2.4:

As a traveler, I want emergency assistance so that I can get help if needed.

- Task 2.4.1: Implement a "SOS Button" to contact emergency services.
- Task 2.4.2: Provide a live chat support system.
- Task 2.4.3: Allow travelers to report safety issues at destinations.

Epic 3: Destination Management for Admins

Feature 5: Destination Management Panel

User Story 3.1:

As an admin, I want to add new destinations so that travelers have more options to explore.

- Task 3.1.1: Design an admin dashboard for adding destinations.
- Task 3.1.2: Implement an API to insert new destination data into the database.
- Task 3.1.3: Ensure verification of destination details before publishing.

User Story 3.2:

As an admin, I want to update destination details (pricing, availability, attractions) so that travelers get accurate information.

- Task 3.2.1: Create an edit destination feature in the admin panel.
- Task 3.2.2: Implement an API for updating destination records.
- Task 3.2.3: Ensure real-time updates reflect on the user interface.

User Story 3.3:

As an admin, I want to remove destinations that are no longer available so that users don’t get outdated information.

- Task 3.3.1: Implement a delete function in the admin dashboard.
- Task 3.3.2: Develop an API to remove destination records.
- Task 3.3.3: Ensure removed destinations don’t affect past bookings or reviews.

Feature 6: User & Content Management

User Story 3.4:

As an admin, I want to manage user accounts so that I can ensure a safe platform.

- Task 3.4.1: Develop user role-based access (Admin, Traveler, Moderator).
- Task 3.4.2: Implement account suspension for violating policies.
- Task 3.4.3: Enable password reset and security measures.

User Story 3.5:

As an admin, I want to monitor reviews and ratings so that I can filter out spam and fake content.

- Task 3.5.1: Develop a review moderation panel.
- Task 3.5.2: Implement AI-based spam detection.
- Task 3.5.3: Allow admins to approve or reject flagged reviews.

ID	User Story	Priority
Epic 1: Destination Selection & Exploration		
US 1.1	Browse available destinations	High
US 1.2	Search and filter destinations	High
US 1.3	View detailed destination information	High
US 1.4	Compare destinations	Medium
Epic 2: Traveler Engagement & Reviews		
US 2.1	Leave reviews and ratings	High
US 2.2	Read traveler reviews and experiences	Medium
US 2.3	Report or moderate reviews	Low
US 2.4	Receive notifications about destination updates	Medium
Epic 3: Destination Management for Admins		
US 3.1	Add new destinations	High
US 3.2	Update destination details	High
US 3.3	Remove outdated destinations	Medium
US 3.4	View & manage user interactions	High
US 3.5	Generate reports & analytics	Medium

Sprint 1: Core Destination Features (2 Weeks)

Goal: Implement the destination browsing and selection system.
Outcome: Users can browse, search, and view destination details.

Sprint 2: Traveler Engagement Features (2 Weeks)

Goal: Enable users to interact with destinations (reviews, recommendations, tracking).
Outcome: Users can leave reviews, receive AI-based destination recommendations, and track their travel locations.

Sprint 3: Destination Management for Admins (2 Weeks)

Goal: Develop the admin panel for destination management.
Outcome: Admins can add, update, or remove destinations and generate reports.

Final Deliverables After Sprint 3

A fully functional Destination Management System with:

- Destination search & selection
- Traveler engagement (reviews, ratings, recommendations, tracking)
- Admin panel for destination & user management
- Reports & analytics for destination performance

Transport Management (IT23210424 – Thivyan J)

Epic 1: Vehicle Booking System

Feature 1: Basic Ride Booking

User Story 1.1:

As a user, I want to browse available Basic Ride vehicles (bike, car, three-wheeler, bus) so that I can select a suitable ride.

- Task 1.1.1: Design and develop the Basic Ride selection UI.
- Task 1.1.2: Implement vehicle listing API to fetch available rides.
- Task 1.1.3: Connect UI with API and display available vehicles.

User Story 1.2:

As a user, I want to book a Basic Ride so that I can travel to my destination.

- Task 1.2.1: Create booking form UI with vehicle selection, date, time, and passenger details.
- Task 1.2.2: Implement backend API for ride booking.
- Task 1.2.3: Integrate payment gateway (if applicable).

Feature 2: Express Ride Booking

User Story 1.3:

As a user, I want to browse available Express Ride vehicles (train, air taxi) so that I can book a faster ride.

- Task 1.3.1: Design the Express Ride selection UI.
- Task 1.3.2: Implement API to fetch train and air taxi availability.
- Task 1.3.3: Display ride options with pricing and schedule.

User Story 1.4:

As a user, I want to book an Express Ride so that I can reach my destination faster.

- Task 1.4.1: Develop booking form and UI for Express Rides.
- Task 1.4.2: Implement backend logic for ride reservation.
- Task 1.4.3: Ensure real-time seat availability updates.

Epic 2: Ride Management for Users

Feature 3: Ride Modification & Cancellation

User Story 2.1:

As a user, I want to cancel my ride so that I can manage my plans.

- **Task 2.1.1**: Develop cancel booking button and UI.
- **Task 2.1.2**: Implement API for ride cancellation.
- **Task 2.1.3**: Handle refund process (if applicable).
-

User Story 2.2: As a user, I want to update my ride booking so that I can modify my travel plans.

  - **Task 2.2.1**: Design modify booking form.
  - **Task 2.2.2**: Implement backend logic to update booking details.
  - **Task 2.2.3**: Notify users of successful modifications.

Feature 4: Ride History & Notifications

User Story 2.3:

As a user, I want to view my past bookings so that I can track my rides.

- Task 2.3.1: Create user booking history UI.
- Task 2.3.2: Develop API to fetch past ride records.
- Task 2.3.3: Display ride details with date, fare, and status.

User Story 2.4:

As a user, I want to receive notifications about my ride status.

- Task 2.4.1: Implement email/SMS notifications for ride confirmation.
- Task 2.4.2: Notify users of cancellations, delays, or changes.
- Task 2.4.3: Display real-time ride status updates in the app.

Epic 3: Transport Management for Admins

Feature 5: Vehicle Management

User Story 3.1:

As a transport manager, I want to add new vehicles so that users have more options.

- Task 3.1.1: Design admin dashboard for vehicle addition.
- Task 3.1.2: Implement API to insert vehicle data into the database.
- Task 3.1.3: Ensure validation of vehicle details.

User Story 3.2:

As a transport manager, I want to update vehicle details (pricing, availability, schedule).

- Task 3.2.1: Create edit vehicle UI in admin panel.
- Task 3.2.2: Develop API to update vehicle records.
- Task 3.2.3: Implement real-time updates in the user interface.

User Story 3.3:

As a transport manager, I want to remove vehicles that are not in service.

- Task 3.3.1: Implement delete vehicle function in the admin dashboard.
- Task 3.3.2: Develop API for removing vehicle records.
- Task 3.3.3: Ensure vehicle removal does not affect ongoing bookings.

Feature 6: Ride Booking Management
User Story 3.4:

As a transport manager, I want to view and manage all ride bookings.

- Task 3.4.1: Create admin dashboard for viewing bookings.
- Task 3.4.2: Implement search and filter options for bookings.
- Task 3.4.3: Allow admin to modify or cancel bookings if required.

Feature 7: Reports & Analytics
User Story 3.5:

As a transport manager, I want to generate reports on vehicle usage.

- Task 3.5.1: Develop report generation feature.
- Task 3.5.2: Implement charts and graphs for data visualization.
- Task 3.5.3: Allow exporting reports as CSV or PDF.

ID	User Story	Priority
Epic 1: Vehicle Booking System		
US 1.1	Browse available Basic Ride vehicles	High
US 1.2	Book a Basic Ride	High
US 1.3	Browse available Express Ride vehicles	High
US 1.4	Book an Express Ride	High
Epic 2: Ride Management for Users		
US 2.1	Cancel a ride	High
US 2.2	Update ride booking	High
US 2.3	View past bookings	Medium
US 2.4	Receive notifications about ride status	Medium
Epic 3: Transport Management for Admins		
US 3.1	Add new vehicles	High
US 3.2	Update vehicle details	High
US 3.3	Remove vehicles	Medium
US 3.4	View & manage ride bookings	High
US 3.5	Generate reports & analytics	Medium

Sprint 1: Core Booking Features (2 weeks)

Goal: Implement the basic and express ride booking system.

Outcome: Users can browse and book basic and express rides.

Sprint 2: Ride Management Features (2 weeks)

Goal: Enable users to manage their rides (cancel, update, history, notifications).
Outcome: Users can modify or cancel rides, view history, and receive alerts.

Sprint 3: Transport Management (2 weeks)

Goal: Develop the admin panel for vehicle and ride management.
Outcome: Admins can manage vehicles, rides, and generate reports.

Final Deliverables After Sprint 3

Fully functional transport management system with:

- Basic & Express Ride booking system
- Ride management (cancel, update, history, notifications)
- Admin panel for vehicle & ride management
- Reports & analytics for transport managers

Hotel Management (IT23210974 – Kesigan M)

Epic 1: Hotel Booking System

Feature 1: Room Search & Booking
User Story 1.1:
As a guest, I want to search and filter available rooms so that I can find the best match based on my preferences.

- Task 1.1.1: Design and develop the room search UI.
- Task 1.1.2: Implement filtering options (price, amenities, availability).
- Task 1.1.3: Connect UI with room availability database.

User Story 1.2:
As a guest, I want to book a room and receive instant confirmation so that I have proof of my reservation.

- Task 1.2.1: Develop room booking form UI.
- Task 1.2.2: Implement booking confirmation via email and SMS.
- Task 1.2.3: Store booking details in the system.

Feature 2: Reservation & Availability Management
User Story 1.3:
As an admin, I want to manage reservations and update room availability so that overbooking is prevented.

- Task 1.3.1: Develop admin panel for reservation management.
- Task 1.3.2: Implement real-time room availability updates.
- Task 1.3.3: Integrate with the booking system.

User Story 1.4:
As an admin, I want to process payments and generate invoices so that guests can complete their transactions securely.

- Task 1.4.1: Set up online payment gateway.
- Task 1.4.2: Develop invoice generation module.
- Task 1.4.3: Implement refund and cancellation policies.

Epic 2: Hotel Management Operations

Feature 3: Pricing & Discounts
User Story 2.1:
As an admin, I want to set pricing for different room types so that I can manage revenue effectively.

- Task 2.1.1: Develop pricing management module.
- Task 2.1.2: Allow admins to update prices.
- Task 2.1.3: Display updated prices to guests.

User Story 2.2:
As an admin, I want to offer discounts and promotions to increase room bookings.

- Task 2.2.1: Implement a discount management system.
- Task 2.2.2: Apply discounts on selected rooms.
- Task 2.2.3: Notify guests about promotions via email/SMS.

Feature 4: Booking Modifications & Cancellations
User Story 2.3:
As a guest, I want to cancel or modify my booking so that I have flexibility in my travel plans.

- Task 2.3.1: Develop booking modification module.
- Task 2.3.2: Implement cancellation policy and refund rules.
- Task 2.3.3: Notify guests about successful modifications.

User Story 2.4:
As an admin, I want to track room occupancy reports so that I can manage availability efficiently.

- Task 2.4.1: Develop occupancy tracking dashboard.
- Task 2.4.2: Generate reports for different time periods.
- Task 2.4.3: Export reports in PDF/Excel format.

Epic 3: Guest Engagement & Notifications

Feature 5: Guest Communication
User Story 3.1:
As an admin, I want to send automated notifications and reminders so that guests receive important updates.

- Task 3.1.1: Implement email/SMS notification system.
- Task 3.1.2: Set up check-in and check-out reminders.
- Task 3.1.3: Notify guests about payment due dates.

User Story 3.2:
As an admin, I want to track guest preferences and feedback so that I can improve customer service.

- Task 3.2.1: Develop guest feedback collection form.
- Task 3.2.2: Store guest preferences in the system.
- Task 3.2.3: Generate guest satisfaction reports

ID	User Story	Priority
Epic 1: Hotel Booking System		
US 1.1	Browse available rooms	High
US 1.2	Book a room	High
US 1.3	Browse available suites	High
US 1.4	Book a suite	High
Epic 2: Reservation & Availability Management		
US 2.1	Manage reservations and update room availability	High
US 2.2	Process payments and generate invoices	High
US 2.3	Set pricing for different room types	Medium
US 2.4	Offer discounts and promotions to increase room bookings	Medium
US 2.5	Track room occupancy reports	Medium
Epic 3: Guest Engagement & Notifications		
US 3.1	Send automated notifications and reminders to guests	Low
US 3.2	Track guest preferences and feedback for improved customer service	Low

Sprint 1: Core Booking Features (2 weeks)

Goal: Implement the basic room booking system.
Outcome: Users can browse available rooms, filter by preferences, and make bookings for different room types (standard rooms, suites).

Sprint 2: Reservation & Availability Management (2 weeks)

Goal: Enable admins to manage reservations, update room availability, and process payments.
Outcome: Admins can manage room reservations, update availability in real-time, process payments, and generate invoices.

Sprint 3: Guest Engagement & Notifications (2 weeks)

Goal: Implement guest communication features, including automated notifications and tracking of guest preferences and feedback.
Outcome: Admins can send automated reminders and notifications to guests, track guest preferences, and collect feedback for service improvement.

Final Deliverables After Sprint 3

A fully functional hotel management system with:

- Basic room booking system (including standard rooms and suites)
- Reservation & availability management system for admins (real-time updates, payment processing, and invoicing)
- Guest engagement features (automated notifications, guest preferences tracking, and feedback collection)

Event Management (IT23158436 – Priyatharshini A H)

EPIC 1: Event Booking System

Feature 1: Basic Event Booking
User Story 1.1
As a user, I want to browse available event venues and event types so that I can select a suitable event for booking.

- Task 1.1.1: Design and develop the Event Selection UI.
- Task 1.1.2: Implement event listing API to fetch available venues and event types.
- Task 1.1.3: Connect UI with API and display available events.

User Story 1.2
As a user, I want to book an event so that I can confirm my participation.

- Task 1.2.1: Create booking form UI with event details, date, time, and number of attendees.
- Task 1.2.2: Implement backend API for event booking.
- Task 1.2.3: Integrate a payment gateway for ticketed events (if applicable).

Feature 2: VIP & Special Event Booking
User Story 1.3
As a user, I want to browse VIP and special events so that I can book premium experiences.

- Task 1.3.1: Design the VIP event selection UI.
- Task 1.3.2: Implement API to fetch VIP event availability.
- Task 1.3.3: Display event details with pricing, available seats, and special offers.

User Story 1.4
As a user, I want to book a VIP or special event so that I can secure my premium seat.

- Task 1.4.1: Develop the booking form and UI for VIP events.
- Task 1.4.2: Implement backend logic for reservation.
- Task 1.4.3: Ensure real-time seat availability updates.

EPIC 2: Event Management for Users

Feature 3: Event Modification & Cancellation
User Story 2.1
As a user, I want to cancel my event booking so that I can manage my plans.

- Task 2.1.1: Develop cancel booking button and UI.
- Task 2.1.2: Implement API for event cancellation.
- Task 2.1.3: Handle refund process (if applicable).

User Story 2.2

As a user, I want to update my event booking so that I can modify my plans.

- Task 2.2.1: Design modify booking form.
- Task 2.2.2: Implement backend logic to update event details.
- Task 2.2.3: Notify users of successful modifications.

Feature 4: Event History & Notifications

User Story 2.3

As a user, I want to view my past bookings so that I can track my event history.

- Task 2.3.1: Create user event history UI.
- Task 2.3.2: Develop API to fetch past event records.
- Task 2.3.3: Display event details with date, location, and status.

User Story 2.4

As a user, I want to receive notifications about my event booking status.

- Task 2.4.1: Implement email/SMS notifications for event confirmation.
- Task 2.4.2: Notify users of cancellations, delays, or changes.
- Task 2.4.3: Display real-time event status updates in the app.

Epic 3: Event Management for Admins

Feature 5: Event Management

User Story 3.1:

As an event manager, I want to add new events so that users have more options to book.

- Task 3.1.1: Design admin dashboard for adding new events.
- Task 3.1.2: Implement API to insert event details into the database.
- Task 3.1.3: Ensure validation of event details (e.g., date, time, capacity).

User Story 3.2:

As an event manager, I want to update event details (pricing, availability, schedule).

- Task 3.2.1: Create edit event UI in the admin panel.
- Task 3.2.2: Develop API to update event records.
- Task 3.2.3: Implement real-time updates in the user interface.

User Story 3.3:

As an event manager, I want to remove events that are no longer available.

- Task 3.3.1: Implement delete event function in the admin dashboard.
- Task 3.3.2: Develop API for removing event records.
- Task 3.3.3: Ensure event removal does not affect ongoing bookings.

Feature 6: Event Booking Management

User Story 3.4:

As an event manager, I want to view and manage all event bookings.

- Task 3.4.1: Create admin dashboard for viewing bookings.
- Task 3.4.2: Implement search and filter options for bookings.
- Task 3.4.3: Allow admin to modify or cancel bookings if required.

Feature 7: Reports & Analytics

User Story 3.5:

As an event manager, I want to generate reports on event bookings and revenue.

- Task 3.5.1: Develop report generation feature.
- Task 3.5.2: Implement charts and graphs for data visualization.
- Task 3.5.3: Allow exporting reports as CSV or PDF.

User Story Id	**User Story**	**Priority**
US01	As a user, I want to book an event so that I can reserve my spot.	High
US02	As a user, I want to cancel my event booking if I change my plan.	High
US03	As an admin, I want to delete an event so that outdated events are removed.	Medium
US04	As a user, I want to view and update my booking details for modifications.	High
US05	As a user, I want to search for events using a search bar.	Medium
US06	As an admin, I want to track event bookings using a CRUD-based system.	High
US07	As a user, I want to leave comments and feedback for events.	Low
US08	As a user, I want to receive notifications for upcoming events.	Medium

Sprint 1 – Core Event Booking System

Outcome: Basic Event Booking System with CRUD operations.

- US01 -Event Booking
- US02 - Cancel Event Booking
- U S04 - View & Update Event Booking

Sprint 2 – Additional Features & User Experience

Outcome: Search functionality, tracking system, and feedback.

- US05 - Search Bar
- US06 - CRUD Tracking System
- US07 - Comments Section

Sprint 3 – Final Enhancements & Notifications

Outcome: Complete system with notifications and full deployment.

- US03 - Delete Event
- US08 - Event Notifications

Final Deliverables After Sprint 3

Fully functional Event Booking Management System with:

- Event Booking System – Users can book events with confirmation.
- Event Management – Cancel, update, delete events, and view booking history.
- Admin Panel – Manage event bookings, track events, and handle deletions.
- Search & Filtering – Users can search for events using keywords.
- User Feedback System – Users can leave comments and reviews.
- Event Notifications – Alerts for upcoming events and booking confirmations.
- Reports & Analytics – Admins can track event statistics and user engagement.

Finance Management (IT23321236 – Nafy F A)

EPIC 1: Financial Transactions Management

Feature 1: Basic Payment Processing

User Story 1.1

As a user, I want to browse available payment options so that I can select a suitable method for my transactions.

- Task 1.1.1: Design and develop the Payment Options UI.
- Task 1.1.2: Implement API to fetch available payment methods.
- Task 1.1.3: Connect UI with API and display available options.

User Story 1.2

As a user, I want to make a payment so that I can complete my transaction.

- Task 1.2.1: Create a payment form UI with transaction details, amount, payment method and QR code.
- Task 1.2.2: Implement backend API for payment processing.
- Task 1.2.3: Integrate a secure payment gateway (e.g., Stripe, PayPal, or local provider).

Feature 2: Refund & Reversal Management

User Story 1.3

As a user, I want to request a refund so that I can get my money back for a failed or canceled transaction.

- Task 1.3.1: Design the refund request UI.
- Task 1.3.2: Implement API to process refund requests.
- Task 1.3.3: Notify users about refund status.

User Story 1.4

As a user, I want to reverse a payment so that I can correct an incorrect transaction.

- Task 1.4.1: Develop the payment reversal request form.
- Task 1.4.2: Implement backend logic for transaction reversals.
- Task 1.4.3: Ensure real-time update of financial records.

EPIC 2: User Financial Management

Feature 3: Transaction History & Statements

User Story 2.1

As a user, I want to view my past transactions so that I can track my expenses.

- Task 2.1.1: Create user transaction history UI.
- Task 2.1.2: Develop API to fetch past transaction records.
- Task 2.1.3: Display transaction details with date, amount, and status.

User Story 2.2
As a user, I want to receive notifications about my financial transactions.

- Task 2.2.1: Implement email/SMS notifications for payment confirmations.
- Task 2.2.2: Notify users of refunds, reversals, or payment failures.
- Task 2.2.3: Display real-time transaction status updates in the app.

Feature 4: Budgeting & Expense Tracking
User Story 2.3

As a user, I want to set a monthly budget so that I can manage my expenses efficiently.

- Task 2.3.1: Develop budget input UI.
- Task 2.3.2: Implement backend logic for tracking expenses against budget.
- Task 2.3.3: Provide visual insights into spending trends.

EPIC 3: Admin Financial Management

Feature 5: Transaction Oversight
User Story 3.1
As a financial manager, I want to add new payment methods so that users have more options.

- Task 3.1.1: Design admin dashboard for adding payment options.
- Task 3.1.2: Implement API to insert new payment methods into the system.
- Task 3.1.3: Ensure validation of payment provider details.

User Story 3.2
As a financial manager, I want to monitor all transactions so that I can ensure compliance and security.

- Task 3.2.1: Create admin panel for transaction oversight.
- Task 3.2.2: Develop API to fetch and monitor transaction logs.
- Task 3.2.3: Implement fraud detection mechanisms.

Feature 6: Refund & Dispute Management
User Story 3.3
As a financial manager, I want to process refund requests efficiently.

- Task 3.3.1: Implement refund approval interface in the admin dashboard.
- Task 3.3.2: Develop API for refund processing.
- Task 3.3.3: Ensure refunds are processed securely and logged.

Feature 7: Financial Reports & Analytics
User Story 3.4
As a financial manager, I want to generate financial reports so that I can track revenue and transactions.

- Task 3.4.1: Develop report generation feature.
- Task 3.4.2: Implement charts and graphs for financial insights.
- Task 3.4.3: Allow exporting reports as CSV or PDF.

User Story IDs & Priorities

ID	User Story	Priority
Epic 1: Financial Transactions Management		
US 1.1	Browse available payment options	High
US 1.2	Make a payment	High
US 1.3	Request for refund	Medium
US 1.4	Reverse a payment	Medium
Epic 2: User Financial Management		
US 2.1	View past transactions	High
US 2.2	Receive financial notifications	High
US 2.3	Set a monthly budget	Low
Epic 3: Admin Financial Management		
US 3.1	Add new payment methods	Medium
US 3.2	Monitor all transactions	High
US 3.3	Process refund requests	Medium
US 3.4	Generate financial reports	Medium

Sprint Plan

Sprint 1 – Core Financial Transactions System
Outcome: Basic Payment Processing with CRUD operations.

- US01 - Payment Processing
- US02 - Cancel Transaction
- US04 - View & Update Transaction History

Sprint 2 – Additional Features & User Experience
Outcome: Search functionality, tracking system, and feedback.

- US05 - Transaction Search
- US06 - CRUD Tracking System
- US07 - Feedback Section

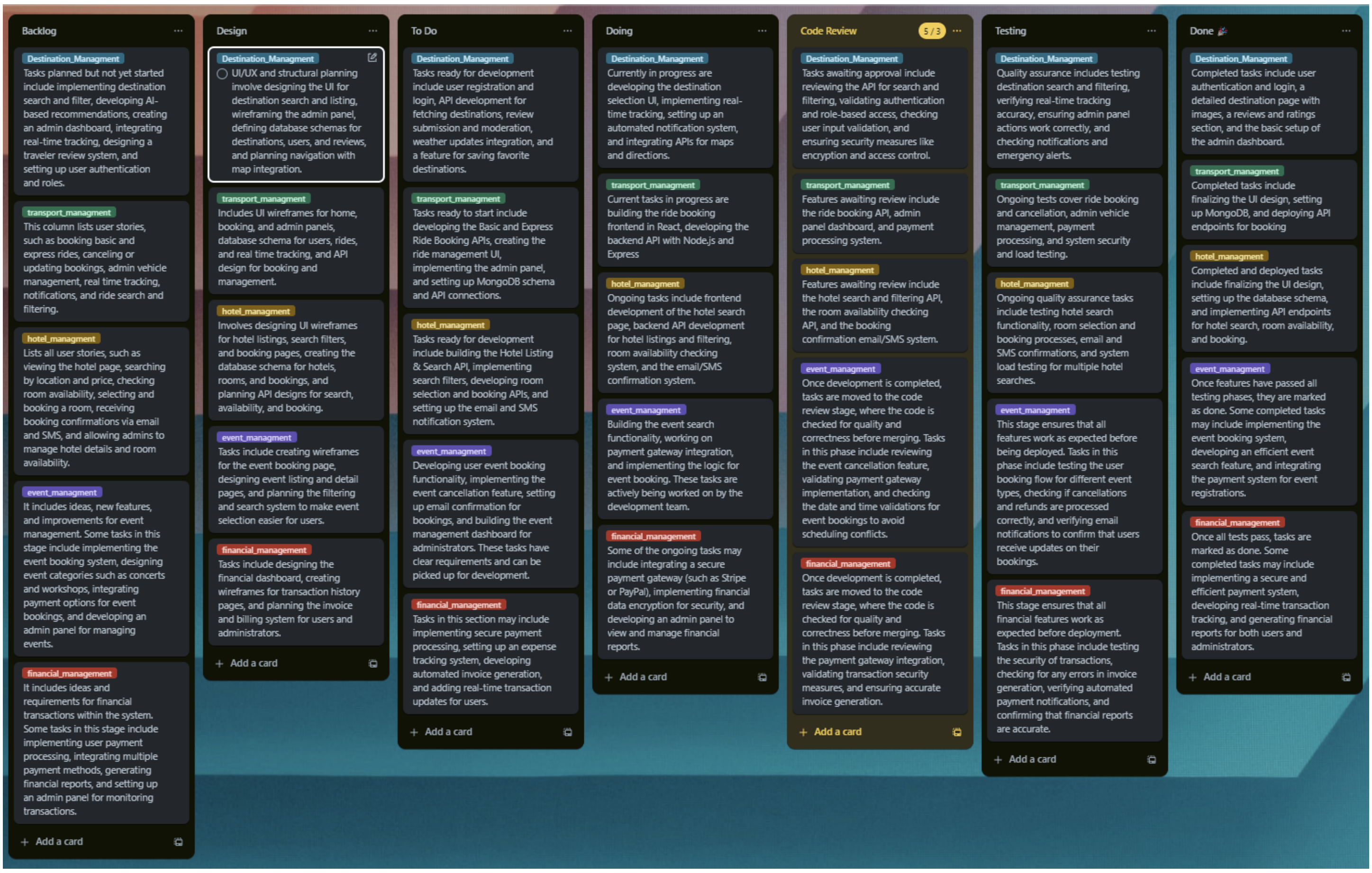
Sprint 3 – Final Enhancements & Notifications
Outcome: Complete system with notifications and full deployment.

- US03 - Remove Payment Options
- US08 - Transaction Notifications

Final Deliverables After Sprint 3

- **Financial Transactions System** – Users can make, cancel, and track payments.
- **Transaction Management** – Users can update, delete transactions, and view history.
- **Admin Panel** – Manage payments, track transactions, and handle disputes.
- **Search & Filtering** – Users can search transactions using filters.
- **User Feedback System** – Users can leave feedback on payment experiences.
- **Notifications** – Alerts for successful payments, refunds, and disputes.
- **Reports & Analytics** – Admins can track financial statistics and generate reports.

Kanban Board



Backlog

Destination_Managment
Tasks planned but not yet started include implementing destination search and filter, developing AI-based recommendations, creating an admin dashboard, integrating real-time tracking, designing a traveler review system, and setting up user authentication and roles.

transport_managment
This column lists user stories, such as booking basic and express rides, canceling or updating bookings, admin vehicle management, real time tracking, notifications, and ride search and filtering.

hotel_managment
Lists all user stories, such as viewing the hotel page, searching by location and price, checking room availability, selecting and booking a room, receiving booking confirmations via email and SMS, and allowing admins to manage hotel details and room availability.

event_managment
It includes ideas, new features, and improvements for event management. Some tasks in this stage include implementing the event booking system, designing event categories such as concerts and workshops, integrating payment options for event bookings, and developing an admin panel for managing events.

financial_management
It includes ideas and requirements for financial transactions within the system. Some tasks in this stage include implementing user payment processing, integrating multiple payment methods, generating financial reports, and setting up an admin panel for monitoring transactions.

Design

Destination_Managment
○ UI/UX and structural planning involve designing the UI for destination search and listing, wireframing the admin panel, defining database schemas for destinations, users, and reviews, and planning navigation with map integration.

transport_managment
Includes UI wireframes for home, booking, and admin panels, database schema for users, rides, and real time tracking, and API design for booking and management.

hotel_managment
Involves designing UI wireframes for hotel listings, search filters, and booking pages, creating the database schema for hotels, rooms, and bookings, and planning API designs for search, availability, and booking.

event_managment
Tasks include creating wireframes for the event booking page, designing event listing and detail pages, and planning the filtering and search system to make event selection easier for users.

financial_management
Tasks include designing the financial dashboard, creating wireframes for transaction history pages, and planning the invoice and billing system for users and administrators.

To Do

Destination_Managment
Tasks ready for development include user registration and login, API development for fetching destinations, review submission and moderation, weather updates integration, and a feature for saving favorite destinations.

transport_managment
Tasks ready to start include developing the Basic and Express Ride Booking APIs, creating the ride management UI, implementing the admin panel, and setting up MongoDB schema and API connections.

hotel_managment
Tasks ready for development include building the Hotel Listing & Search API, implementing search filters, developing room selection and booking APIs, and setting up the email and SMS notification system.

event_managment
Developing user event booking functionality, implementing the event cancellation feature, setting up email confirmation for bookings, and building the event management dashboard for administrators. These tasks have clear requirements and can be picked up for development.

financial_management
Tasks in this section may include implementing secure payment processing, setting up an expense tracking system, developing automated invoice generation, and adding real-time transaction updates for users.

Doing

Destination_Managment
Currently in progress are developing the destination selection UI, implementing real-time tracking, setting up an automated notification system, and integrating APIs for maps and directions.

transport_managment
Current tasks in progress are building the ride booking frontend in React, developing the backend API with Node.js and Express

hotel_managment
Ongoing tasks include frontend development of the hotel search page, backend API development for hotel listings and filtering, room availability checking system, and the email/SMS confirmation system.

event_managment
Building the event search functionality, working on payment gateway integration, and implementing the logic for event booking. These tasks are actively being worked on by the development team.

financial_management
Some of the ongoing tasks may include integrating a secure payment gateway (such as Stripe or PayPal), implementing financial data encryption for security, and developing an admin panel to view and manage financial reports.

Code Review

Destination_Managment
Tasks awaiting approval include reviewing the API for search and filtering, validating authentication and role-based access, checking user input validation, and ensuring security measures like encryption and access control.

transport_managment
Features awaiting review include the ride booking API, admin panel dashboard, and payment processing system.

hotel_managment
Features awaiting review include the hotel search and filtering API, the room availability checking API, and the booking confirmation email/SMS system.

event_managment
Once development is completed, tasks are moved to the code review stage, where the code is checked for quality and correctness before merging. Tasks in this phase include reviewing the event cancellation feature, validating payment gateway implementation, and checking the date and time validations for event bookings to avoid scheduling conflicts.

financial_management
Once development is completed, tasks are moved to the code review stage, where the code is checked for quality and correctness before merging. Tasks in this phase include reviewing the payment gateway integration, validating transaction security measures, and ensuring accurate invoice generation.

Testing

Destination_Managment
Quality assurance includes testing destination search and filtering, verifying real-time tracking accuracy, ensuring admin panel actions work correctly, and checking notifications and emergency alerts.

transport_managment
Ongoing tests cover ride booking and cancellation, admin vehicle management, payment processing, and system security and load testing.

hotel_managment
Ongoing quality assurance tasks include testing hotel search functionality, room selection and booking processes, email and SMS confirmations, and system load testing for multiple hotel searches.

event_managment
This stage ensures that all features work as expected before being deployed. Tasks in this phase include testing the user booking flow for different event types, checking if cancellations and refunds are processed correctly, and verifying email notifications to confirm that users receive updates on their bookings.

financial_management
This stage ensures that all financial features work as expected before deployment. Tasks in this phase include testing the security of transactions, checking for any errors in invoice generation, verifying automated payment notifications, and confirming that financial reports are accurate.

Done 🎉

Destination_Managment
Completed tasks include user authentication and login, a detailed destination page with images, a reviews and ratings section, and the basic setup of the admin dashboard.

transport_managment
Completed tasks include finalizing the UI design, setting up MongoDB, and deploying API endpoints for booking

hotel_managment
Completed and deployed tasks include finalizing the UI design, setting up the database schema, and implementing API endpoints for hotel search, room availability, and booking.

event_managment
Once features have passed all testing phases, they are marked as done. Some completed tasks may include implementing the event booking system, developing an efficient event search feature, and integrating the payment system for event registrations.

financial_management
Once all tests pass, tasks are marked as done. Some completed tasks may include implementing a secure and efficient payment system, developing real-time transaction tracking, and generating financial reports for both users and administrators.

Trello Kanban Board Link - Activity 03_ITP25_B1_C1-92.docx