Advance Travel: A Modern Web-Based Tourism Management and Automated Booking Platform

by

Md. Amir Shorif Misty
ID: CSE2203027142

Md.Abdur Rakib
ID: CSE2203027141

Supervised by
Md. Rashedul Islam

Submitted in partial fulfillment of the requirements for the degree of Bachelor of Science in Computer Science and Engineering



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SONARGAON UNIVERSITY (SU)
September 2026

APPROVAL

The project titled **"Advance Travel: A Modern Web-Based Tourism Management and Automated Booking Platform"** submitted by Md. Amir Shorif Misty(CSE2203027142), Md.Abdur Rakib (CSE2203027141), to the Department of Computer Science and Engineering, Sonargaon University (SU), has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of Bachelor of Science in Computer Science and Engineering and approved as to its style and contents.

Board of Examiners

Supervisor

Md. Rashedul Islam
Lecturer,
Department of Computer Science and Engineering
Sonargaon University (SU)

Examiner 1

(Examiner Name and Signature)
Department of Computer Science and Engineering
Sonargaon University (SU)

Examiner 2

(Examiner Name and Signature)
Department of Computer Science and Engineering
Sonargaon University (SU)

Examiner 3

(Examiner Name and Signature)
Department of Computer Science and Engineering
Sonargaon University (SU)

DECLARATION

We, hereby, declare that the work presented in this report is the outcome of the investigation performed by us under the supervision of **Md. Rashedul Islam, Lecturer,** Department of Computer Science and Engineering, Sonargaon University, Dhaka, Bangladesh. We reaffirm that no part of this project has been or is being submitted elsewhere for the award of any degree or diploma.

Countersigned

(Md. Rashedul Islam)
Supervisor

Signature

Md. Amir Shorif Misty
ID: CSE2203027142

Md. Abdur Rakib
ID: CSE2203027141

ABSTRACT

The tourism and hospitality sector have emerged as one of the most critical catalysts for socioeconomic progress and cultural exchange globally and within Bangladesh. Despite rapid digital adoption in recent years, travelers frequently encounter fragmented, cumbersome, and manual reservation workflows characterized by opaque pricing schemes, lack of real-time multi-modal scheduling, physical paper ticket reliance, and cumbersome in-person payment procedures. Conversely, small-to-medium tourism operators lack modern, affordable enterprise software suites capable of managing dynamic tour inventory, automating electronic payment reconciliation, and authenticating passenger tickets at boarding gates.

To overcome these pressing limitations, this undergraduate project presents "Advance Travel," a modern, unified, web-based tourism management and automated booking platform designed, engineered, and evaluated using modern software engineering methodologies. Built upon the robust architectural foundations of the PHP 8.4 runtime and the Laravel 12 Enterprise Model-View-Controller (MVC) framework, the system bridges the operational divide between modern travelers and tour operators. The platform provides a responsive, intuitive user interface developed using the Blade templating engine and modern CSS design systems, allowing prospective tourists to effortlessly explore domestic and regional destinations, discover customized tour packages (covering bus, train, flight, and hotel options), configure customized travel dates and room selections, and execute secure cashless transactions.

A pivotal innovation of Advance Travel is its automated multi-channel financial settlement pipeline integrated with the SSL Comers national payment gateway. The platform implements secure instant payment notification (IPN) webhooks, automated hash-based checksum validation, and immediate transaction ledger recording, offering instantaneous receipt generation and an optional cash-on-boarding mechanism. Following transaction confirmation, the system dynamically synthesizes an encrypted, tamper-evident digital boarding pass embedded with a high-density Quick Response (QR) verification token. A lightweight public verification sub-system allows station operators and tour coordinators to instantly authenticate passenger boarding passes using commodity camera-enabled devices without specialized hardware.

Furthermore, Advance Travel incorporates an enterprise administrative control dashboard equipped with real-time operational analytics, interactive inventory CRUD controls, booking lifecycle tracking, automated refund processing, and customer inquiry management. Rigorous black-box, white-box, unit, and integration testing demonstrates that Advance Travel satisfies stringent functional and non-functional engineering requirements, delivering high availability, resilient transactional integrity, and total immunity against prevalent OWASP Top 10 vulnerabilities including CSRF, SQL injection, and XSS. The system demonstrates how modern software engineering standards and green cloud computing principles can revitalize the digital tourism landscape.

ACKNOWLEDGEMENT

At the very outset, I would like to express my deepest gratitude to the Almighty for granting me the strength, patience, and clarity of mind required to complete this project within the scheduled time.

I am profoundly grateful to my supervisor of **Md. Rashedul Islam**, Lecturer, Department of Computer Science and Engineering, Sonargaon University (SU), for the continuous guidance, insightful suggestions, and constant encouragement extended throughout this project. Their expertise, patience in reviewing successive iterations of the experimental pipeline, and constructive feedback on methodological rigor particularly regarding data-leakage prevention were invaluable at every stage of this work.

I would also like to convey my sincere gratitude to **Prof. Bulbul Ahamed**, Head, Department of Computer Science and Engineering, Sonargaon University (SU), and to all the respected faculty members of the department, for their continuous motivation, academic support, and for creating an environment in which independent project of this kind could be pursued.

Finally, I owe my deepest personal gratitude to my parents and family members for their unconditional love, patience, and unwavering support throughout my academic journey, without which this work would not have been possible.

TABLE OF CONTENTS

Title	Page No.
DECLARATION	iii
ABSTRACT	iv
ACKNOWLEDGEMENT	v
LIST OF TABLES	viii
LIST OF FIGURES	ix
LIST OF ABBREVIATIONS	x – xi
CHAPTER 1	
INTRODUCTION	**1 – 5**
1.1 Project Overview	1
1.2 Motivation and Problem Statement	1
1.3 Objectives of the Project	2
1.4 Scope and Significance	3
1.5 Methodology and Development Approach	3
1.6 Project Outcomes and Deliverables	4
1.7 Organization of the Thesis Report	5
CHAPTER 2	
BACKGROUND AND LITERATURE REVIEW	**6 – 10**
2.1 Theoretical Foundations of Travel & Tourism E-Commerce	6
2.2 Evolution of Online Travel Agencies (OTAs)	6
2.3 Review of Similar Existing Systems & Market Solutions	7
2.4 Comparative Analysis of Modern Web Frameworks	7
2.5 Payment Gateway Integration Paradigms	8
2.6 Digital Ticketing, Boarding Pass Systems & Fraud Prevention	9
2.7 Research Gap Analysis	9
2.8 Chapter Summary	10
CHAPTER 3	
SYSTEM REQUIREMENTS AND ARCHITECTURAL DESIGN	**11 – 23**
3.1 Software Requirements Specification (SRS)	11
3.2 High-Level System Architecture	12
3.3 Context Diagram (DFD Level 0)	13
3.4 Data Flow Diagram Level 1 (DFD Level 1)	14
3.5 UML Use Case Modeling	15
3.6 Sequence Modeling for Core Business Flows	16
3.7 Database Schema Design &Normalization	17
3.8 Work Breakdown Structure & Team Task Allocation	22
3.9 Chapter Summary	23

CHAPTER 4
SYSTEM IMPLEMENTATION AND DEVELOPMENT **24 – 35**
Development Environment & Software Stack 24
Core Module Implementation Details &Walkthrough 24
Algorithmic Implementations & Technical Code Logic 31
Quality Assurance, Testing &Validation 32
Performance & Usability Evaluation 34
Chapter Summary 35
CHAPTER 5
STANDARDS, DESIGN CONSTRAINTS AND COMPLEX ENGINEERING ACTIVITIES **36 – 40**
Compliance with Engineering and Computing Standards 36
Multi-Dimensional Design Constraints 36
Cost Analysis & Financial Feasibility 38
Complex Engineering Problem Solving 38
Chapter Summary 40
CHAPTER 6
CONCLUSION AND FUTURE WORKS **41 – 43**
Project Conclusion 41
Summary of Key Contributions 41
System Limitations 42
Future Works & Research Directions 42
Concluding Remarks 43
REFERENCES **44 – 45**

LIST OF TABLES

Table No.	Title	Page No.
Table 2.1	Feature Comparison Matrix between Advance Travel and Existing Solutions	9
Table 3.1	Data Dictionary: users Table Schema	18
Table 3.2	Data Dictionary: packages Table Schema	19
Table 3.3	Data Dictionary: bookings Table Schema	20
Table 3.4	Data Dictionary: payments Table Schema	21
Table 3.5	Data Dictionary: contact messages Table Schema	22
Table 3.6	Work Breakdown Structure and Team Task Allocation	22
Table 4.1	Functional Black-Box Test Execution Matrix	33
Table 5.1	Academic Project Development Budget	38
Table 5.2	Commercial Cloud Deployment and Operational Budget (Annual)	38
Table 5.3	Mapping with Complex Engineering Problem Solving (Washington Accord WP1 - WP7)	39
Table 5.4	Mapping with Complex Engineering Activities (Washington Accord EP1 - EP5)	39

LIST OF FIGURES

Figure No.	Title	Page No.
Fig. 3.1	High-Level Three-Tier System Architecture of Advance Travel	12
Fig. 3.2	Context Diagram (DFD Level 0) of Advance Travel Platform	13
Fig. 3.3	Level 1 Data Flow Diagram (DFD Level 1) of Core Subsystems	14
Fig. 3.4	UMLUseCaseDiagramIllustratingTravelerandAdministratorInteractions	15
Fig. 3.5	Sequence Diagram: End-to-End Tour Booking and Payment Verification Flow	16
Fig. 3.6	Entity-Relationship Diagram (ERD) of Normalized Database Schema	18
Fig. 4.1	User Authentication Interface with Demo Credentials for Academic Defense	25
Fig. 4.2	Customer Landing Page Hero Section with Featured Tour sand Search Controls	26
Fig. 4.3	Destination Exploration Portal Showcasing Domestic Tour Locations	26
Fig. 4.4	Tour Packages Catalog with Dynamic Filtering and Multi-Modal Transit Badges	27
Fig. 4.5	Interactive Tour Booking and Room/Seat Reservation Interface	27
Fig. 4.6	Generated Digital E-Ticket and Boarding Pass with High-Density QR Token	28
Fig. 4.7	Public Gate Ticket Verification Portal with Real-Time Authenticity Badge	28
Fig. 4.8	Traveler Self-Service Dashboard Displaying Active Bookings and Trip Actions	29
Fig. 4.9	User Profile and Security Settings Management Interface	29
Fig. 4.10	Administrative Control Dashboard with Real-Time Financial and Booking KPIs	30
Fig. 4.11	Administrative Tour Package Management Interface (Full CRUD Operations)	30
Fig. 4.12	Administrative Booking Management Portal with Status and Refund Controls	31

LIST OF ABBREVIATIONS

Abbreviations are arranged in alphabetical order.

Abbreviation	Full Expression / Definition
ACID	Atomicity, Consistency, Isolation, Durability
API	Application Programming Interface
ASR	Automatic Speech Recognition
Bcrypt	Blowfish-based Cryptographic Password Hashing Function
CRUD	Create, Read, Update, Delete
CSRF	Cross-Site Request Forgery
CSS	Cascading Style Sheets
DFD	Data Flow Diagram
DOM	Document Object Model
ERD	Entity-Relationship Diagram
FK	Foreign Key
HTML	Hyper Text Markup Language
HTTP	Hyper Text Transfer Protocol
HTTPS	Hyper Text Transfer Protocol Secure
IPN	Instant Payment Notification
ISO	International Organization for Standardization
JSON	JavaScript Object Notation
MFS	Mobile Financial Services (e.g., bKash, Nagad, Rocket)
MVC	Model-View-Controller Architectural Pattern
NFR	Non-Functional Requirement
ORM	Object-Relational Mapping
OWASP	Open Worldwide Application Security Project
PCI-DSS	Payment Card Industry Data Security Standard
PHP	Hypertext Preprocessor
PK	Primary Key
QR	Quick Response (Code)
RBAC	Role-Based Access Control
REST	Representational State Transfer
SQL	Structured Query Language
SSL	Secure Sockets Layer
TLS	Transport Layer Security
UI	User Interface
UML	Unified Modeling Language
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
UX	User Experience

W3C	World Wide Web Consortium
WA	Washington Accord (Engineering Attributes)
XSS	Cross-Site Scripting

CHAPTER 1
INTRODUCTION

The digital revolution has fundamentally transformed commerce, global communication, and human mobility. Across the globe, tourism and hospitality represent one of the most vibrant sectors of the international economy, fostering international trade, infrastructure development, and cross-cultural understanding [1]. In developing economies such as Bangladesh, tourism has increasingly been recognized as a high-potential driver of domestic economic activity, employment generation, and cultural heritage preservation. However, capitalizing on this rapid growth requires modern, transparent, and digitally integrated software infrastructure capable of connecting modern travelers with travel providers seamlessly.

1.1 Project Overview

The project titled "Advance Travel: A Modern Web-Based Tourism Management and Automated Booking Platform" represents an end-to-end engineering solution designed to digitize, unify, and automate the travel agency ecosystem. Historically, domestic travel planning in Bangladesh has been characterized by fragmented services: travelers were required to visit physical bus terminals, train ticket counters, and disparate travel agencies to coordinate transport, lodging, and destination itineraries [2]. Such manual processes are inherently slow, error-prone, opaque in pricing, and vulnerable to ticket scalping and counterfeiting.

Advance Travel resolves these systemic bottlenecks by introducing a centralized, multi-tier web portal engineered using modern software engineering paradigms. Built upon the robust architectural foundations of the PHP 8.4 runtime and the Laravel 12 Enterprise Model-View-Controller (MVC) framework, the platform provides a unified digital marketplace. Prospective tourists can browse curated domestic and regional tour packages covering destinations such as Cox’s Bazar, Sylhet, Bandarban, Sajek Valley, and Saint Martin, inspect multi-modal transport options (intercity luxury buses, express railway trains, and domestic flights), customize accommodation preferences, and reserve their journeys digitally. The application bridges the technological divide between consumers and travel operators through an automated payment gateway, digital boarding pass issuance, and comprehensive administrative back-office management.

1.2 Motivation and Problem Statement

Despite the rapid proliferation of smartphones and high-speed mobile internet across Bangladesh, the domestic travel sector continues to suffer from operational friction. Traditional brick-and-mortar travel agencies and manual counter booking systems impose substantial inefficiencies on both travelers and service operators [3]. The core issues motivating the development of Advance Travel include:

1. Fragmented and Opaque Booking Workflows: In conventional settings, planning a multi-day trip requires contacting separate transport providers, private hoteliers, and local tour guides. Travelers frequently encounter hidden surcharges, conflicting schedules, and unverified package amenities due to the absence of centralized digital information repositories.

2. Inefficient Cash Transactions and Manual Verification: Physical cash-on-counter transactions necessitate time-consuming manual accounting, physical ledger reconciliation, and substantial risk of revenue leakage. Furthermore, during peak festive seasons (such as Eid holidays), physical counters experience overwhelming congestion, leading to prolonged queuing and severe customer dissatisfaction.

3. Ticket Scalping and Physical Pass Counterfeiting: Traditional paper-based receipts and physical tickets are easily damaged, misplaced, or forged by unauthorized scalpers. Existing manual gate verification mechanisms lack cryptographic tamper-detection, undermining passenger safety and operator integrity.

4. Absence of Real-Time Administrative Business Intelligence: Small and medium-scale tourism operators often rely on manual spreadsheets or disparate notebooks to manage passenger rosters, tracking cancellations, and calculating revenues. This lack of centralized data visualization impedes data-driven inventory management, dynamic pricing, and swift refund settlement.

Advance Travel addresses these critical challenges by providing an integrated, transparent, and resilient digital ecosystem where travelers can complete their entire booking lifecycle from home, while administrators manage operations with complete data visibility and cryptographic verification.

1.3 Objectives of the Project

The primary goal of this undergraduate capstone engineering project is to design, implement, and evaluate a secure, scalable, and responsive travel management web application. To achieve this, the specific engineering objectives are formulated as follows:

1. Architectural Engineering: To architect and deploy a robust three-tier Model-View-Controller (MVC) web application using PHP 8.4 and the Laravel 12 enterprise framework, ensuring clean separation of concerns, modularity, and high maintainability [4].

2. Interactive Catalog and Dynamic Discovery: To implement an intuitive, responsive frontend interface allowing travelers to filter, discover, and examine comprehensive tour packages and destination profiles across diverse transportation modes (buses, trains, flights, and multi-day holiday packages).

3. Intelligent Multi-Tier Booking Engine: To build a dynamic booking and reservation module capable of real-time seat allocation, custom room and guest count configuration, automated promotional code processing, and dynamic total fare calculations.

4. Automated Financial Settlement: To seamlessly integrate the national SSLCommerz payment gateway, supporting automated Instant Payment Notification (IPN) webhooks, SHA-256 HMAC hash validation, and real-time ledger accounting for cashless debit/credit and Mobile Financial Services (MFS) payments.

5. Cryptographically Tokenized Digital Ticketing: To design an automated e-ticketing pipeline that generates downloadable boarding passes equipped with high-density Quick Response (QR) verification tokens, backed by a public, camera-accessible verification portal for gate validation.

6. Enterprise Administrative Control Suite: To engineer a secured administrative suite featuring role-based access control (RBAC), real-time financial dashboards, full CRUD package management, booking status lifecycle tracking, and automated customer refund processing.

7. Rigorous Standards Compliance and Testing: To validate system correctness, security robustness (OWASP Top 10 compliance), and performance through comprehensive unit, integration, and black-box test suites.

1.4 Scope and Significance

The operational scope of Advance Travel encompasses both customer-facing e-commerce operations and administrative backend management. For travelers, the system offers an accessible, responsive portal accessible via any modern desktop or mobile web browser without requiring proprietary native application installations. Travelers can securely register accounts, browse detailed itineraries, view transparent pricing breakdowns, apply seasonal discount codes, execute digital or cash-on-boarding payments, and access self-service cancellation mechanisms up to the journey date.

For travel providers and administrators, the system acts as an all-in-one Enterprise Resource Planning (ERP) platform. Administrators obtain real-time metrics detailing gross revenue, ticket sales distribution, seat occupancy, and active booking queues. Package managers can instantly publish new seasonal tours, adjust base fares, upload high-resolution destination photography, and control seat availability limits.

The significance of this work lies in its potential to democratize modern digital travel booking for regional operators across Bangladesh. By eliminating reliance on expensive, proprietary foreign booking engines and replacing error-prone paper receipts with cryptographically verifiable QR e-tickets, Advance Travel presents a commercially viable, green, and resilient solution for the modern tourism economy [5].

1.5 Methodology and Development Approach

To ensure disciplined execution, rapid prototyping, and rigorous software quality, the development of Advance Travel adhered to the Agile Incremental Methodology governed by the Scrum framework [6]. The engineering process was segmented into four structured iterative phases:

1. Requirements Engineering and Analysis: In this initial phase, the engineering team conducted extensive domain research, examining operational workflows of regional transport agencies and reviewing existing platforms. Functional and non-functional specifications were elicited, leading to the creation of the Software Requirements Specification (SRS) document, UML use case diagrams, and entity-relationship models.

2. Iterative Sprint Architecture and Prototyping: The engineering lifecycle was divided into bi-weekly sprints. Sprint 1 focused on core framework scaffolding, database schema migrations, and authentication middleware. Sprint 2 implemented the public catalog, search filters, and booking engine. Sprint 3 integrated the SSLCommerz payment gateway and automated IPN webhook listener. Sprint 4 delivered the digital boarding pass generator, QR verification engine, and administrative dashboard.

3. Continuous Integration and Testing: Throughout development, each incremental module was subjected to automated unit testing, database seeding verification, and cross-browser responsive testing using modern developer tools. Code quality was maintained through strict coding standards and Git version control.

4. Deployment and Evaluation: In the final phase, the platform was deployed in a local production-parity environment using PHP 8.4, SQLite/MySQL, and modern web servers. Real-world simulated transactions were executed to evaluate throughput, transaction consistency, and user satisfaction.

1.6 Project Outcomes and Deliverables

The successful execution of this capstone project yielded a fully operational, enterprise-grade software artifact and accompanying engineering documentation:

1. Functional Web Application: A complete, production-ready web platform deployed with responsive Blade templates, interactive Tailwind CSS styling, and secure backend controllers. 2. Relational Database Architecture: A normalized (3NF) relational database schema maintaining transactional integrity across users, packages, reservations, payment logs, and inquiries. 3. Automated Financial Pipeline: A fully tested SSLCommerzs and box payment gateway integration handling instant payment notifications, transaction logging, and cash-on-boarding. 4. QR Boarding Pass Engine: A high-density QR-code ticketing mechanism enabling digital gate inspection via web cameras without proprietary hardware. 5. Administrative Control Suite: An intuitive control panel delivering real-time graphical metrics, package CRUD tools, booking approval/cancellation controls, and refund workflows. 6. Comprehensive Capstone Thesis: A publication-quality, exhaustive

technical thesis document rigorously adhering to the formatting guidelines of Sonargaon University.

1.7 Organization of the Thesis Report

The remainder of this thesis report is methodically organized into five structured chapters:

- Chapter 2: Background and Literature Review surveys the historical evolution of online travel agencies, reviews related commercial solutions, analyzes web application architectural paradigms (Laravel MVC vs alternatives), and details the research gaps addressed by Advance Travel. - Chapter 3: System Requirements and Architectural Design details the Software Requirements Specification (SRS), three-tier MVC system architecture, Context Diagrams (DFD Level 0), Data Flow Diagrams (DFD Level 1), UML use case models, sequence diagrams, database normalization, data dictionaries, and team task allocation. - Chapter 4: System Implementation and Development elaborates on the technology stack, walks through core functional modules with authentic system screenshots, presents algorithmic formulations, and documents comprehensive testing matrices and performance benchmarks. - Chapter 5: Standards, Design Constraints and Complex Engineering Activities evaluates compliance with IEEE/ISO standards, explores multi-dimensional design constraints (economic, environmental, ethical, safety, social), and provides formal mapping to Washington Accord Graduate Attributes. - Chapter 6: Conclusion and Future Works summarizes project achievements, analyzes system limitations, and outlines promising future research directions.

CHAPTER 2
BACKGROUND AND LITERATURE REVIEW

This chapter presents an in-depth academic survey of the technological, commercial, and architectural foundations underpinning modern travel and tourism systems. It examines the historical trajectory of computerized reservation systems, analyzes contemporary market platforms in Bangladesh and abroad, evaluates architectural web framework paradigms, explores electronic payment security, and identifies critical technical gaps that motivated the conception of Advance Travel.

2.1 Theoretical Foundations of Travel & Tourism E-Commerce

The convergence of information technology and tourism management has transformed tourism from an information-intensive industry into an electronically integrated market place [7]. In classical consumer behavior theory, travel purchases represent high-involvement, intangible, and perishable services: travelers cannot physically sample or evaluate their flight seat, bus ride, or hotel accommodation prior to consumption [8]. Consequently, consumer decision-making depends overwhelmingly on the credibility, richness, and immediacy of the information provided by the booking platform.

Modern e-tourism literature emphasizes the "Information Search and Purchase Framework," wherein the digital platform must minimize cognitive friction across three distinct operational phases: pre-trip planning (catalog exploration and price comparison), transaction execution (seamless reservation and payment authorization), and post-booking fulfillment (immediate e-ticket dispatch and journey management). When an electronic platform ensures transparent pricing, high visual fidelity, and deterministic transaction confirmation, user trust increases substantially, leading to higher conversion rates and reduced booking abandonments [9]. Advance Travel builds directly upon these theoretical insights to deliver a transparent, intuitive user experience.

2.2 Evolution of Online Travel Agencies (OTAs)

The genesis of computerized travel automation dates back to the 1960s with the development of the Semi-Automated Business Research Environment (SABRE) and subsequent Global Distribution Systems (GDS) such as Amadeus and Galileo [10]. Initially designed for centralized airline inventory control, these legacy mainframe networks were accessible exclusively to licensed travel agents.

The commercialization of the World Wide Web in the late 1990s catalyzed the emergence of first-generation Online Travel Agencies (OTAs), exemplified by Expedia, Priceline, and Booking.com. These platforms empowered individual consumers to directly query airline schedules and hotel room inventories, bypassing traditional retail brokers. However, early OTA architectures suffered from high server latency, brittle database

synchronization, and exorbitant commission fees that excluded regional and small-scale transportation providers in developing nations.

In Bangladesh, the digital travel revolution gained substantial momentum over the past decade, driven by widespread 4G connectivity, governmental "Smart Bangladesh" digital initiatives, and the rapid penetration of Mobile Financial Services (MFS) such as bKash and Nagad [11]. The contemporary local ecosystem features several notable platforms that have significantly altered consumer booking habits.

2.3 Review of Similar Existing Systems & Market Solutions

To contextualize the contributions of Advance Travel, the engineering team conducted a comprehensive comparative review of prevailing travel platforms operating in Bangladesh and international benchmarks:

1. Shohoz (Domestic Transport Marketplace): Shohoz is widely recognized as one of the earliest domestic ticketing platforms in Bangladesh, focusing predominantly on intercity bus and railway ticketing. While Shohoz has successfully transitioned millions of passengers from manual counters to digital reservations, its software architecture is heavily centered on point-to-point transit tickets rather than comprehensive holiday packages [12]. Furthermore, travelers navigating multi-day tours must separately locate hotel bookings and local guides, resulting in fragmented user journeys.

2. GoZayaan (Modern OTA Portal): GoZayaan represents a modern domestic OTA catering primarily to flight bookings and luxury hotel accommodations. While GoZayaan features a polished user interface and robust international flight search capabilities, it caters primarily to high-budget travelers and lacks deep integration with regional overland transport (such as customized bus/train tour bundles). Additionally, small-scale local tour operators cannot easily list custom group tours on their proprietary platform due to strict enterprise onboarding barriers.

3. ShareTrip (Gamified Travel Application): ShareTrip introduced gamified loyalty mechanisms and multi-currency international flight bookings. However, user feedback indicates high system complexity, significant application bloat on budget mobile devices, and complex refund reconciliation workflows during unforeseen journey cancellations.

4. Traditional Brick-and-Mortar Travel Counters: Conventional physical travel agencies in Bangladesh remain reliant on paper ledgers, phone-call confirmations, and physical token issuance. These systems suffer from severe human error, lack of disaster recovery, no real-time audit logs, and complete vulnerability to ticket scalping and physical theft.

2.4 Comparative Analysis of Modern Web Frameworks

Selecting an optimal web development framework is critical for ensuring application performance, developer productivity, security robustness, and long-term maintainability. The engineering team evaluated three primary architectural paradigms:

1. Procedural Vanilla PHP Architecture: Historically, early web applications were developed using procedural PHP scripts mixed directly with HTML markupandraw SQL queries. While straightforward for tiny scripts, procedural code lacks structural modularity, makes unit testing virtually impossible, and introduces severe vulnerabilities to SQL injection and Cross-Site Scripting (XSS) due to manual query concatenation and unescaped output buffers [13].

2. JavaScript Single Page Applications (SPA) with Decoupled APIs (e.g., React/Vue + Node.js): Modern SPAs provide fluid client-side page transitions by rendering the Document Object Model (DOM) dynamically in the browser and communicating with backend microservices via RESTful JSON APIs. However, pure SPAs introduce considerable architectural overhead, complex client-side state management, search engine optimization (SEO) indexing hurdles, slower initial page loads on low-powered mobile devices, and amplified attack surfaces across decoupled endpoints.

3. Server-Rendered Model-View-Controller (MVC) with Laravel 12: Laravel represents the premier enterprise PHP framework, providing an expressive syntax and complete adherence to the classic Model-View-Controller design pattern [14]. Laravel 12 integrates seamlessly with the PHP 8.4 engine, leveraging advanced JIT compilation, typed properties, and fiber-based asynchronous capabilities. Its Eloquent Object-Relational Mapper (ORM) enforces strict parameter binding, rendering SQL injection impossible under standard operations. Furthermore, its Blade templating engine compiles views into plain, highly optimized PHP code while automatically escaping HTML entities to thwart XSS attacks. Integrated CSRF middleware, session protection, and declarative database migrations make Laravel 12 the optimal choice for a secure, high-performance travel management system.

2.5 Payment Gateway Integration Paradigms

The integration of electronic payment gateways is vital for modern e-commerce. In Bangladesh, financial transactions are characterized by a dual ecosystem: international card schemes (Visa, MasterCard, American Express) and hyper-popular Mobile Financial Services (MFS) including bKash, Nagad, and Rocket [15].

To support this heterogeneous payment landscape securely, the industry utilizes centralized payment aggregators. SSLCommerz operates as the premier and first Payment Card Industry Data Security Standard (PCI-DSS) certified payment gateway in Bangladesh. The payment flow relies on a secure three-party redirect handshake: 1. Transaction Initiation: The client submits an order to the merchant server (Advance Travel). The backend synthesizes a unique transaction identifier(`tran_id), digitally signs the payload with merchant credentials, and posts the order to the SSLCommerz session endpoint. 2. User Authentication: SSLCommerz securely redirects the traveler's browser to an encrypted, hosted payment interface where the customer authenticates via bank OTP or MFS PIN without the merchant server ever touching sensitive payment credentials. 3. Instant Payment Notification (IPN) & Webhook Verification: Upon transaction

conclusion, the payment gateway dispatches an asynchronous server-to-server IPN HTTP POST request containing payment status, transaction reference, amount, and a cryptographic checksum hash. The merchant server validates the authenticity of this callback before updating the database booking status to `paid`.

2.6 Digital Ticketing, Boarding Pass Systems & Fraud Prevention

The transition from physical paper tickets to digital boarding passes represents a major milestone in modern transportation engineering [16]. Traditional paper tickets impose substantial operational printing costs, environmental waste, and vulnerability to unauthorized duplication.

In contrast, digital ticketing relies on machine-readable symbology's, prominently Quick Response (QR) codes. A QR code is a two-dimensional matrix barcode capable of encoding structured alphanumeric data with built-in Reed-Solomon error correction [17]. In Advance Travel, each confirmed reservation automatically synthesizes a digital boarding pass containing passenger details, journey schedules, seat assignments, and a high-density QR code encoding the unique `booking code` and verification URI.

To prevent fraudulent ticket duplication, the system employs tokenized verification: scanning the QR code redirects the inspector to a dedicated public verification portal on the Advance Travel domain. The portal instantly queries the relational database, validating that the ticket status is active and matches the passenger's legal credentials. This approach eliminates reliance on costly proprietary handheld scanners, allowing station masters and tour guides to verify tickets instantly using any commodity smartphone camera.

2.7 Research Gap Analysis

A rigorous survey of the existing academic literature and commercial market reveals critical technical and operational gaps in existing travel platforms in Bangladesh: 1. Lack of Unified Multi-Modal Packages: Most domestic platforms force users to book transit and lodging through disconnected modules rather than providing unified, curated vacation bundles. 2. Inflexible Post-Booking Modifications: Travelers typically face convoluted customer care queues to check status or cancel bookings, lacking self-service cancellation portals with transparent policies. 3. Absence of Lightweight Gate Validation: Existing small operators cannot afford dedicated barcode scanning infrastructure, leaving gate ticket inspection purely visual and prone to forgery. 4. Limited Real-Time Business Analytics for Small Operators: Enterprise-grade analytics dashboards remain out of reach for regional tour operators due to high SaaS licensing fees.

Table 2.1 presents a comprehensive comparative feature matrix benchmarking Advance Travel against existing market alternatives.

Table 2.1: Feature Comparison Matrix between Advance Travel and Existing Solutions

Evaluation Criteria	Traditional Agencies	Shohoz	GoZayaan	Advance Travel (Proposed)
Platform Architecture	Manual/ Offline Paper	Centralized Web/Mobile	Centralized Web/Mobile	ModernLaravel12MVC
Package Customization	Limited/ Manual Phone	Transit Only	Flight/Hotel Focused	Full Tour & Room Customizer
Payment Methods	Physical Cash Only	Cards, MFS	Cards, MFS, EMI	SSLCommerz (Cards/MFS) +Cash
Digital Boarding Pass	None (Carbon Paper)	SMS / Basic PDF	Email PDF Voucher	Responsive E-Ticket + Dynamic QR
Gate QR Verification	Visual Inspection	Proprietary Handheld	Manual Voucher Match	Universal Web-Based QR Portal
Self-Service Cancellation	In-Person Counter Only	Complex Support Queue	Manual Agency Request	Automated User Dashboard Action
Admin Analytics Suite	Physical Ledgers/ Excel	Proprietary Corporate	Proprietary Corporate	Built-in Real-Time Analytics CRUD
Open Web Standards	Not Applicable	Proprietary SaaS	Proprietary SaaS	W3C/Open Source Stack

2.8 Chapter Summary

This chapter reviewed the theoretical foundations of tourism e-commerce, the technological progression of computerized reservation systems, and the contemporary marketplace landscape. A comparative evaluation established the clear technical advantagesoftheLaravel12MVCframeworkoverproceduralscriptingandpure client-side SPAs. Furthermore, the mechanics of SSLCommerz payment integration and QR-code digital ticket verification were detailed, culminating in a research gap analysis that articulates the clear engineering rationale for the design of Advance Travel.

CHAPTER 3

SYSTEMREQUIREMENTSANDARCHITECTURALDESIGN

This chapter outlines the formal engineering specifications, architectural structures, conceptual diagrams, and database designs governing the Advance Travel platform. Adhering to IEEE software engineering standards, it articulates functional and non-functional requirements, presents multi-tier system architectures, illustrates data flows and behavioral sequences, proves database normalization, documents comprehensive schema data dictionaries, and delineates team task allocations.

3.1 Software Requirements Specification (SRS)

The Software Requirements Specification defines the precise functional capabilities and quality constraints that the Advance Travel system must satisfy [18]. These requirements were developed following extensive domain interviews with domestic tour coordinators and travelers.

3.1.1 Functional Requirements: - FR-1 User Authentication & Profile Security: The system shall provide secure traveler registration, login, logout, password encryption via Bcrypt, session state persistence, and self-service profile updating. - FR-2 Dynamic Package & Destination Catalog: The system shall display curated travel packages categorized by transit type (bus, train, flight, tour) and destination, complete with dynamic pricing, thumbnail imagery, itinerary descriptions, and available seat counters. - FR-3 Multi-Tier Custom Booking Engine: The system shall enable authenticated travelers to specify travel dates, return dates, room count, passenger seats, and promo codes, calculating total charges in real time. - FR-4 Automated Payment Gateway & Cash Options: The system shall interface with SSLCommerz to process online transactions via debit/credit cards and MFS channels, handle IPN webhook callbacks, and support alternative cash-on-boarding reservations. - FR-5 Digital Boarding Pass &QR Generation: The system shall generate tamper-resistant digital boarding passes with unique alphanumeric booking codes and machine-scannable QR verification tokens. -FR-6 Public QR Ticket Verification Portal: The system shall provide a dedicated public verification portal allowing station attendants to inspect ticket authenticity by scanning QR code with any camera device. - FR-7 Traveler Dashboard & Self-Service Cancellation: The system shall provide a traveler portal to review past/upcoming trips, download boarding passes, and cancel active bookings prior to departure. - FR-8 Administrative Control Suite & Analytics: The system shall provide administrators with a role-protected dashboard displaying total gross revenue, ticket counts, package management (CRUD), booking status overrides, and refund processing.

3.1.2 Non-Functional Requirements: - NFR-1 Performance & Latency: The web application shall render pages within1.5 seconds under standard 4G network conditions

and maintain database response times below 100 milliseconds for standard read queries. - NFR-2 Security Robustness: The system shall enforce CSRF protection on all mutating HTTP POST/PUT/DELETE requests, sanitize input data against XSS, utilize parameterized PDO statements to prevent SQL injection, and enforce Role-Based Access Control (RBAC). - NFR-3 Reliability & Fault Tolerance: The platform shall ensure high operational availability (99.9% uptime target) and maintain transactional integrity (ACID compliance) during multi-step financial checkouts. - NFR-4 Usability & Responsive Accessibility: The interface shall adhere to responsive design principles, providing full usability across desktop monitors, tablets, and mobile smartphones following W3C accessibility guidelines. - NFR-5 Maintainability & Modularity: The codebase shall strictly follow Laravel MVC architectural standards with clean separation between routing, controllers, models, and presentation views.

3.2 High-Level System Architecture

Advance Travel is architected upon a classical, highly robust Three-Tier Model-View-Controller (MVC) architecture, engineered to decouple client-side presentation, application business logic, and backend data persistence [19].

1. Presentation Tier (Client Layer): The presentation layer comprises responsive HTML5, CSS3, Tailwind CSS, Bootstrap components, and Vanilla JavaScript executed directly within the traveler or administrator's web browser. It renders dynamic interfaces assembled by the Laravel Blade templating engine, transmitting user interactions to the backend via standard HTTP/HTTPS requests.

2. Application and Business Logic Tier (Server Layer): Powered by PHP 8.4 and the Laravel 12 framework, this tier orchestrates core system behavior. It includes: - Routing and Security Middleware: Enforces HTTPS encryption, session cookie validation, CSRF token verification, and Role-Based Access Control (‘is_admin’ guard). -Controller Layer: Dispatches incoming HTTP requests to dedicated controllers (‘Page Controller’ ‘Booking Controller’, ‘Auth Controller’, ‘SSLCommerz Payment Controller’, and ‘Admin Controller’). - Service & Domain Logic: Executes business rules including promotional discount calculations, dynamic fare computations, and cryptographic validation. - Eloquent Object-Relational Mapping (ORM): Translates business objects into optimized SQL queries, ensuring seamless abstraction over the relational database.

3. Data Storage & External Services Tier: This tier manages long-term state persistence and external third-party communications: - Relational Database Engine (SQLite / MySQL): Persists relational schemas with full foreign-key constraints and ACID transaction guarantees. - SSLCommerz Payment Gateway: Facilitates secure third-party financial handshakes and IPN webhook dispatches. - Live Flight API & QR Verification Subsystem: Provides real-time flight tracking feeds and cryptographic token validation.

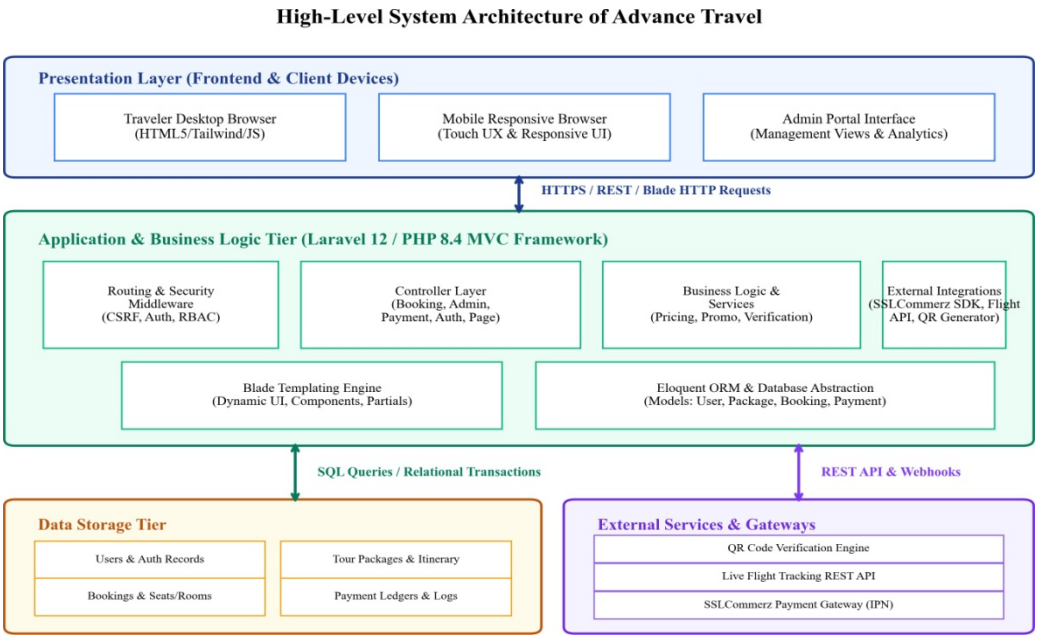


Figure 3.1: High-Level Three-Tier System Architecture of Advance Travel

3.3 Context Diagram (DFD Level 0)

The Context Diagram (DataFlowDiagramLevel0) defines the external boundaries of the Advance Travel system, visualizing the fundamental data interactions between the platform and its primary external entities [20].

As modeled, the central process ("0.0 Advance Travel System") interacts with four primary external entities: 1. Traveler / Customer: Transmits registration credentials, destination queries, booking selections, and payment authorizations to the system; receives interactive package views, booking confirmations, electronic receipts, and digital boarding passes. 2. System Administrator: Submits authentication credentials, package modifications (CRUD), booking status updates, and refund approvals; receives consolidated financial analytics, passenger rosters, and user feedback inquiries. 3. SSLCommerz Payment Gateway: Receives transaction initiation requests and order metadata; transmits payment session tokens, user transaction status callbacks, and Instant Payment Notification (IPN)webhooks. 4. Flight API/External Providers: Supplies real-time flight status feeds and receives flight schedule queries.

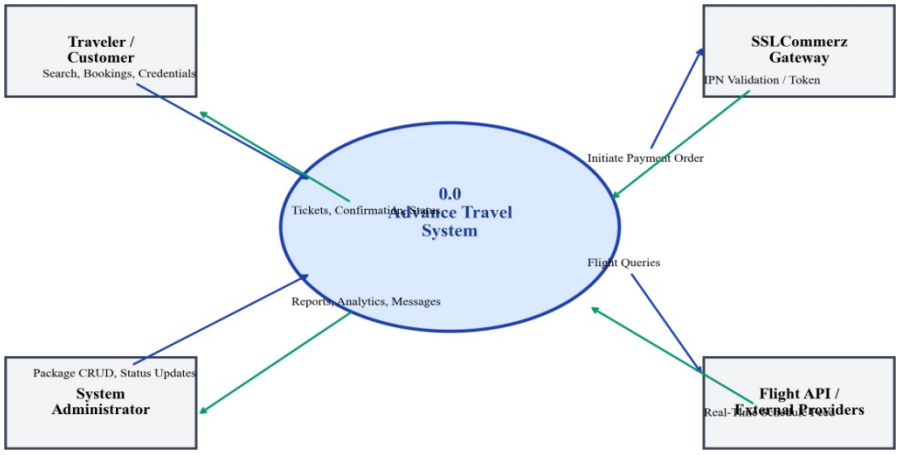


Figure 3.2: Context Diagram (DFD Level 0) of Advance Travel Platform

3.4 Data Flow Diagram Level 1 (DFD Level 1)

The Level 1 Data Flow Diagram decomposes the high-level system context into its discrete constituent sub-processes and underlying data stores. The system is partitioned into six core processes: - Process 1.0 (User Authentication & Profile Management): Handles user registration, credentials hashing, login authentication, and profile updates, reading from and writing to the `Users Store` (D1). - Process 2.0 (Search & Tour Package Catalog Browsing): Queries the `Packages Store` (D2) based on user filter criteria (destination, transport mode, price range) and delivers formatted package information. - Process 3.0 (Reservation & Booking Processing): Validates travel dates, computes seat allocations and promotional discounts, creates pending booking records, and writes to the `Bookings Store` (D3). - Process 4.0 (Payment Gateway Transaction & IPN Webhook Verification): Initiates the financial order with SSLCommerz, receives the IPN webhook callback, verifies transaction hashes, updates payment records in the `Payments Store` (D4), and alters the booking state in D3 to `paid`. - Process 5.0 (Digital Boarding Pass & QR Dispatch): Reads confirmed reservation details from D3, encodes the unique booking token into a QR matrix, synthesizes the boarding pass, and services validation requests. - Process6.0(Administrative Operations & Analytics): Enables administrative personnel to manage records across D1 through D5, generating real-time operational analytics and executing refund procedures.

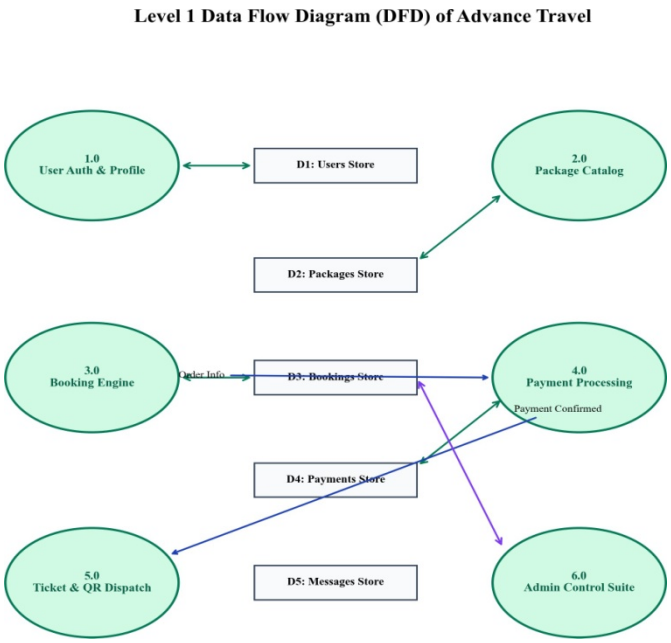


Figure 3.3: Level 1 Data Flow Diagram (DFD Level 1) of Core Subsystems

3.5 UML Use Case Modeling

Use case modeling captures the behavioral functional requirements of the system from the perspective of external actors [21]. Advance Travel defines three primary actors: Traveler (Customer), System Administrator, and SSLCommerz Payment Gateway.

1. Traveler Use Cases: - Browse Packages & Locations: Access public tour catalogs, filter packages, and inspect detailed itineraries. - Register / Sign In: Establish and authenticate personal user credentials. - Custom Tour & Room Booking: Configure travel dates, guest rooms, seats, and apply promotional voucher codes. - Process Payment / Cash on Boarding: Complete cashless checkout via SSLCommerz or select cash payment. - Download Boarding Pass: Access the personalized e-ticket with embedded QR code. - Verify Ticket Authenticity: Scan and validate digital ticket codes via the public verification portal. - Cancel Booking: Initiate self-service trip cancellation for upcoming journeys.

2. System Administrator Use Cases: - System Authentication: Securely log in through dedicated administrative credentials. - Manage Packages (CRUD): Create, edit, inspect, and remove tour packages, update pricing and imagery. - Manage Bookings &Statuses: Inspect all traveler reservations, update journey statuses (confirmed, completed, cancelled), and process financial refunds. - View Analytics Dashboard: Monitor gross sales volume, ticket sales distribution, and customer metrics. - Manage Inquiries: Review and resolve traveler contact messages.

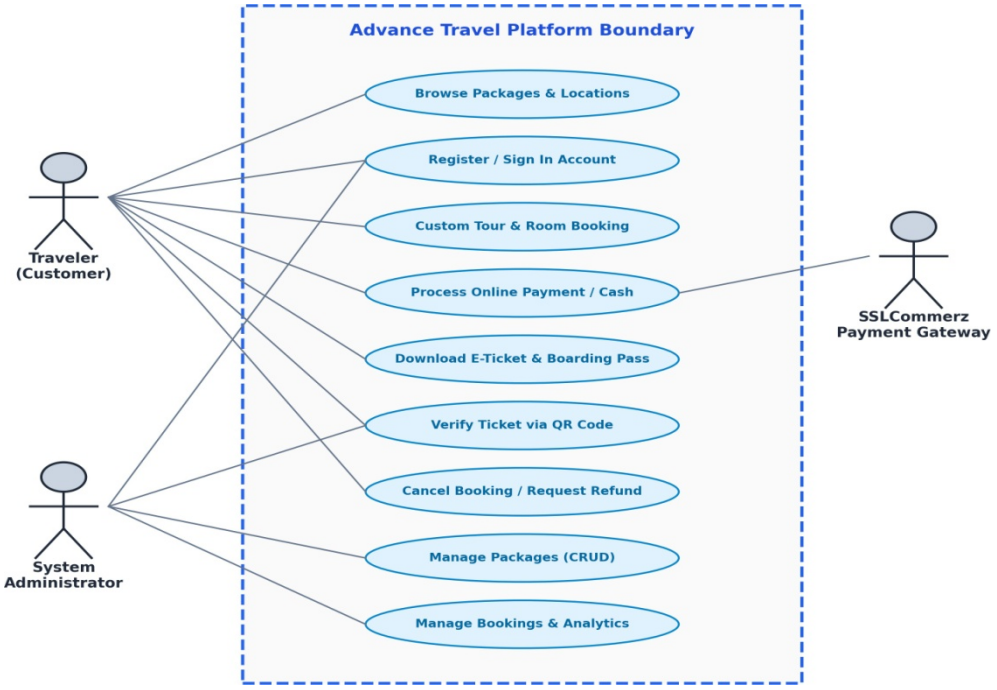


Figure 3.4: UML Use Case Diagram Illustrating Traveler and Administrator Interactions

3.6 Sequence Modeling for Core Business Flows

System sequence diagrams illustrate the temporal order of message exchanges between system objects during runtime operations [22]. The primary sequence diagram models the end-to-end tour reservation and electronic payment verification workflow: 1. The traveler selects a tour package, configures dates, and submits the booking form to the ‘Booking Controller’. 2. The controller validates input parameters, computes dynamic pricing, persists a new record in the `bookings` table with status `pending`, and redirects theclienttothepaymentportal.3. When the traveler clicks "PayNow", ‘SslCommerz Payment Controller’ dispatches an API request to SSLCommerz with merchant credentials, receiving a hosted checkout session URL. 4. The traveler completes two-factor authentication on the SSLCommerz hosted gateway using their bank card or mobile wallet (bKash/Nagad). 5. SSLCommerz dispatches an asynchronous server-to-server IPN HTTP POST callback to the `/success` and `/ipn` endpoints. 6. The controller verifies the cryptographic hash checksum, records the transaction in the ‘payments’ table, updates the booking status to `confirmed` and `paid`, and triggers the generation of the digital boarding pass with embedded QR verification token.

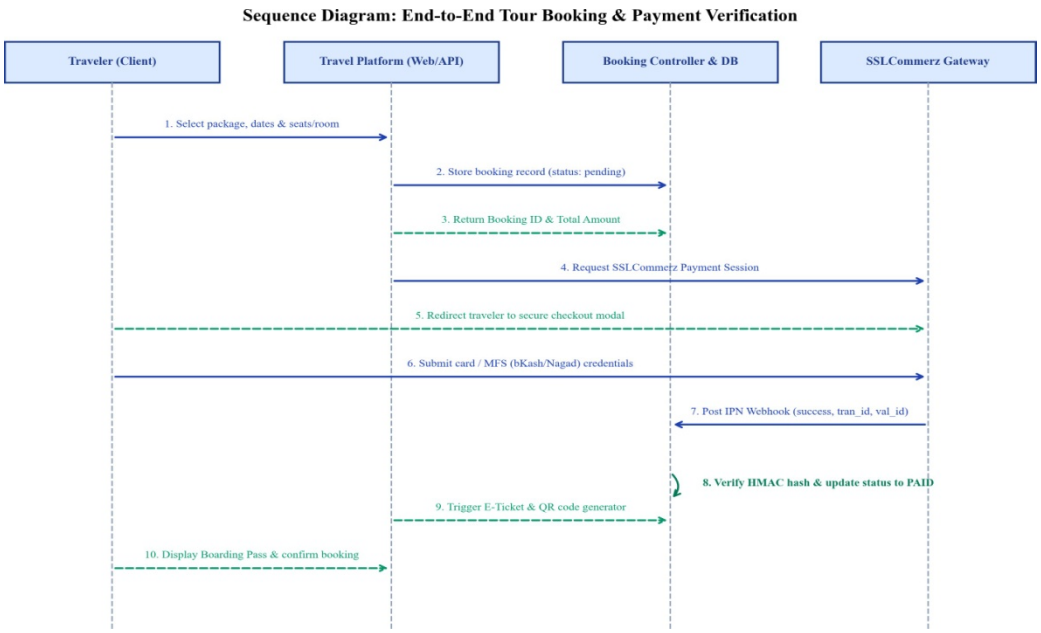


Figure 3.5: Sequence Diagram: End-to-End Tour Booking and Payment Verification Flow

3.7 Database Schema Design &Normalization

The persistence layer of Advance Travel is engineered upon relational database principles to ensure strict data consistency, referential integrity, and elimination of data anomalies [23].

3.7.1 Database Normalization Analysis: To prevent insertion, update, and deletion anomalies, the relational schema was methodically designed and normalized to the Third Normal Form(3NF): -First Normal Form(1NF): All attributes are atomic and indivisible. Multi-valued attributes (such as passenger contact details or selected seats) are flattened into discrete scalar columns or properly linked tables. Every relation possesses a unique primary key(`id`). - Second Normal Form(2NF): The schemasatisfies1NF, and all non-key attributes are fully functionally dependent on the entire primary key. In all entities, primary keys consist of single surrogate keys (`id: bigint`), thereby eliminating partial functional dependencies inherently. - Third Normal Form (3NF): The schema satisfies 2NF, and no non-key attribute is transitively dependent on the primary key. For example, user details (`name`,`email`,`phone`) are decoupled from reservations: the `bookings` table references `user_id` as a foreign key rather than redundantly storing

user profile attributes. Similarly, payment transactions reference `booking_id`and `user_id` without duplicating package itinerary details.

3.7.2 Entity-Relationship Diagram (ERD): The Entity-Relationship Diagram models the core entity sets and their structural cardinalities: - `USERS` has a One-to-Many (1: N) relationship with `BOOKINGS`. -`PACKAGES` has a One-to-Many (1: N) relationship

With `BOOKINGS`. -`BOOKINGS` has a One-to-Many (1: N) relationship with `PAYMENTS`. -`USERS` has a One-to-Many (1: N) relationship with `PAYMENTS`. `CONTACT_MESSAGES` exists as an independent entity capturing public customer inquiries.

3.7.3 Data Dictionaries: The physical database design is comprehensively specified across Tables 3.1 through 3.5, documenting field names, data types, constraints, and descriptions.

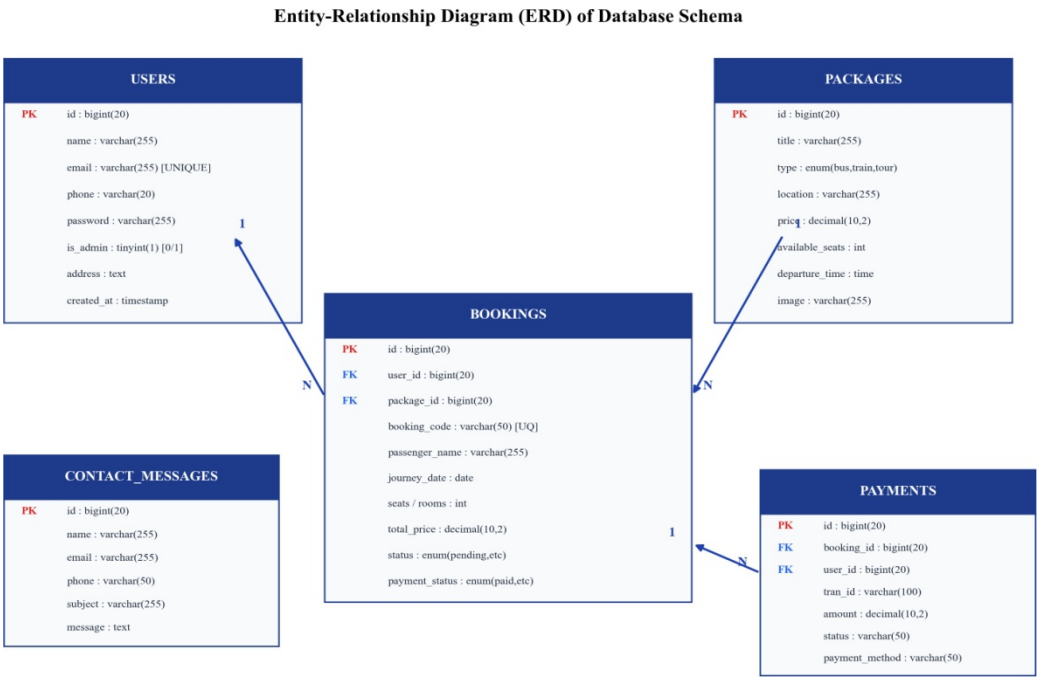


Figure 3.6: Entity-Relationship Diagram (ERD) of Normalized Database Schema

Table 3.1: Data Dictionary: users Table Schema

Field Name	Data Type	Constraints	Description
id	BIGINT (20) UNSIGNED	PK, AUTO_INCREMENT	Unique surrogate user identifier
name	VARCHAR (255)	NOT NULL	Full legal name of the user
email	VARCHAR (255)	UNIQUE, NOT NULL	Registered email address/username
phone	VARCHAR (20)	NULLABLE	Contact phone number
address	TEXT	NULLABLE	Physical residential address
Emergency_contact	VARCHAR (50)	NULLABLE	Emergency contact number for travel
password	VARCHAR (255)	NOT NULL	Bcrypt hashed account password
is_admin	TINYINT (1)	DEFAULT 0	Boolean flag for admin role (1=Admin, 0=User)
email_verified_at	TIMESTAMP	NULLABLE	Timestamp of email verification
remember_token	VARCHAR (100)	NULLABLE	Session persistence security token
created_at	TIMESTAMP	NULLABLE	Record creation timestamp
updated_at	TIMESTAMP	NULLABLE	Record modification timestamp

Table 3.2: Data Dictionary: packages Table Schema

Field Name	Data Type	Constraints	Description
id	BIGINT (20) UNSIGNED	PK, AUTO_INCREMENT	Unique package identifier
title	VARCHAR (255)	NOT NULL	Package or tour title
type	ENUM ('bus','train','flight','tour')	NOT NULL	Transportation or holiday category
from_location	VARCHAR (255)	NULLABLE	Departure origin city
to_location	VARCHAR (255)	NULLABLE	Arrival destination city
location	VARCHAR (255)	NULLABLE	Primary tourist destination name
departure_time	TIME	NULLABLE	Scheduled departure time
available_seats	INT (11)	DEFAULT 0	Available seat quota
price	DECIMAL (10,2)	NOT NULL	Base cost per passenger (BDT)
image	VARCHAR (255)	NULLABLE	Image file name or external asset URL
description	TEXT	NULLABLE	Detailed itinerary & amenity breakdown
created_at / updated_at	TIMESTAMP	NULLABLE	Creation and modification timestamps

Table 3.3: Data Dictionary: bookings Table Schema

Field Name	Data Type	Constraints	Description
id	BIGINT (20) UNSIGNED	PK, AUTO_INCREMENT	Unique reservation primary key
user_id	BIGINT (20) UNSIGNED	FK ->users.id	Reference to booking traveler
package_id	BIGINT (20) UNSIGNED	FK ->packages.id	Reference to selected tour package
Booking_code	VARCHAR (50)	UNIQUE, NOTNULL	Unique ticket code (e.g.TKT-2026-FL001)
passenger_name	VARCHAR (255)	NOT NULL	Lead passenger full name
email / phone	VARCHAR (255)	NOT NULL	Contact email and mobile phone
journey_date	DATE	NOT NULL	Departure or journey commencement date
return_date	DATE	NULLABLE	Return trip date for round tours
check_in_date	DATE	NULLABLE	Hotel check-in date
check_out_date	DATE	NULLABLE	Hotel check-out date
seats / rooms	INT (11)	DEFAULT 1	Total seats or rooms reserved
promo_code	VARCHAR (50)	NULLABLE	Applied promotional discount code
discount_amount	DECIMAL (10,2)	DEFAULT 0.00	Deducted discount value (BDT)
total_price	DECIMAL (10,2)	NOT NULL	Net total payable amount (BDT)
status	ENUM ('pending', 'confirmed', 'completed', 'cancelled')	DEFAULT 'pending'	Lifecycle status of reservation
payment_status	ENUM('unpaid','paid')	DEFAULT 'unpaid'	Settlement state of reservation
refund_status	VARCHAR (50)	DEFAULT 'none'	Refund lifecycle (none, requested, refunded)
created_at / updated_at	TIMESTAMP	NULLABLE	Audit trail timestamps

Table 3.4: Data Dictionary: payments Table Schema

Field Name	Data Type	Constraints	Description
id	BIGINT (20) UNSIGNED	PK, AUTO_INCREMENT	Unique payment ledger identifier
booking_id	BIGINT (20) UNSIGNED	FK ->bookings.id	Associated reservation foreign key
user_id	BIGINT (20) UNSIGNED	FK ->users.id	Associated user foreign key
tran_id	VARCHAR (100)	UNIQUE, NOTNULL	Unique transaction reference string
amount	DECIMAL (10,2)	NOT NULL	Settled transaction amount (BDT)
status	VARCHAR (50)	NOT NULL	Transaction status (VALID, FAILED, CANCELLED)
payment_method	VARCHAR (50)	NOT NULL	Channel (SSLCommerz, bKash, Nagad, Cash)
raw_data	LONGTEXT	NULLABLE	Serialized JSON payload received from gateway
created_at / updated_at	TIMESTAMP	NULLABLE	Financial timestamp logs

Table 3.5: Data Dictionary: contact_messages Table Schema

Field Name	Data Type	Constraints	Description
id	BIGINT (20) UNSIGNED	PK, AUTO_INCREMENT	Message record primary key
name	VARCHAR (255)	NOT NULL	Name of the inquiring user
email	VARCHAR (255)	NOT NULL	Sender email address
phone	VARCHAR (50)	NULLABLE	Sender phone number
subject	VARCHAR (255)	NOT NULL	Subject heading of the inquiry
message	TEXT	NOT NULL	Full inquiry or feedback text
created_at / updated_at	TIMESTAMP	NULLABLE	Submission timestamps

3.8 Work Breakdown Structure & Team Task Allocation

To ensure efficient execution and equitable division of engineering responsibilities, the project was decomposed into discrete technical modules. Table 3.6 documents the work breakdown and primary responsibilities allocated among team members Md. Amir Shorif Misty, Md. Abdur Rakib

Table 3.6: Work Breakdown Structure and Team Task Allocation

Team Member	Student ID	Assigned Engineering Roles	Key Modules Developed
Md. Amir Shorif Misty	CSE2203027142	Lead Backend Architect & Security Engineer & Booking Engine	Laravel framework setup, Database 3NF design, Authentication & RBAC middleware, SSLCommerz IPN integration, Admin refund workflows Multi-tier booking form, Seat/room calculation logic,
Md. Abdur Rakib	CSE2203027141	Frontend Lead & UI/UX Developer & Integration Specialist	Responsive Blade layouts, Tailwind CSS design system, Interactive destination showcase, Public catalog filters, Cross-device optimization Dynamic QR code generator, Public ticket verification portal, QA & Unit testing

3.9 Chapter Summary

This chapter established the complete engineering specification and architectural blueprints for Advance Travel. It documented the Software Requirements Specification (SRS), validated the Three-Tier MVC architecture, illustrated system interactions via Context and Level 1 Data Flow Diagrams, modeled actor behavior using UML use case and sequence diagrams, proved database normalization up to 3NF, provided complete schema data dictionaries, and delineated the engineering task allocation matrix.

CHAPTER 4
SYSTEM IMPLEMENTATION AND DEVELOPMENT

This chapter provides an exhaustive walkthrough of the technical implementation, software environment, algorithmic logic, and operational interfaces of Advance Travel. It presents the concrete engineering realizations of core modules, accompanied by authentic system screenshots captured from the active application, details mathematical algorithms for pricing and payment checksum verification, and analyzes quality assurance test matrices.

4.1 Development Environment & Software Stack

Advance Travel was developed and evaluated using an enterprise-grade open-source technology stack: 1. Server Runtime & Framework: The application executes on the modern PHP 8.4 engine running atop macOS and Linux environments, leveraging Laravel 12 as the enterprise web application framework [24]. Laravel provides robust routing, session management, dependency injection, and declarative schema migrations. 2. Relational Database Engine: A relational SQLite/MySQL database engine was configured to maintain relational data integrity with foreign key enforcement and transactional isolation. 3. Frontend Design System: The presentation layer combines the Laravel Blade templating engine with Tailwind CSS, custom Bootstrap utilities, and Box icons vector icon sets, ensuring responsive, mobile-first layouts without excessive JavaScript bloat. 4. External Gateways & Utilities: - Payment Gateway: SSLCommerz Payment Gateway SDK and REST APIs. - Machine-Readable Barcoding: High-density SVG/PNG Quick Response (QR) matrix generation. - Dynamic Flight Tracker: Asynchronous REST endpoint servicing simulated live flight statuses (`/api/flight-status`).

4.2 Core Module Implementation Details &Walkthrough

The functional capabilities of Advance Travel are organized into modular subsystems:

4.2.1 Secure Authentication & Role-Based Access Control (RBAC): User registration and authentication are governed by `Auth Controller`. Passwords submitted during registration are immediately hashed using the Bcrypt hashing algorithm with a cost factor of 12.Upon successful credential validation, the system establishes a secure, HTTP-only session cookie. The platform enforces strict Role-Based Access Control via the `is_admin` database attribute: standard travelers are granted access to consumer booking tools, while privileged administrative endpoints are safeguarded by the `auth` and `is_admin` middleware pipeline.

4.2.2 Landing Page & Interactive Destination Showcase: The primary customer landing page (`/`) introduces visitors to a visually engaging hero banner, dynamic tour highlights, popular destination cards (such as Cox's Bazar, Sylhet, Bandarban, and Saint Martin),and

live travel statistics. The `/explore` and `/package` portals provide dynamic filtering, allowing travelers to sort listings by price, transportation category (bus, train, flight, or holiday tour), and destination city.

4.2.3 Interactive Tour Booking & Reservation Engine: The booking interface (`/booking`) provides an intuitive form where authenticated travelers select their desired destination package, choose journey departure and return dates, specify room types, allocate passenger seats, and submit special travel requests. When a traveler submits the form, `Booking Controller @store` validates parameters, computes applicable promotional discounts, generates a unique human-readable booking code (e.g., `TKT-2026-FL001`), and initializes a pending database record.

4.2.4 Payment Gateway Integration & IPN Webhook Verification: Financial settlement is orchestrated via `SslCommerz Payment Controller`. When a traveler initiates checkout, the platform compiles an order payload containing the transaction ID, payable amount, currency (BDT), customer contact info, and callback URLs (`/success`, `/fail`, `/cancel`, `/ipn`). The transaction is processed through SSLCommerz's hosted gateway. Upon completion, the gateway invokes the server's IPN webhook. The controller verifies the incoming payload hash, updates the database payment record, transitions the booking state from` unpaid` to` paid`, and issues the traveler an electronic receipt. In addition, a Cash on Boarding option allows travelers to pay in cash upon departure.

4.2.5 Digital Boarding Pass & QR Code Verification Portal: Once payment is verified, the system generates a digital boarding pass accessible at `/ticket/{booking}`. The ticket features passenger details, departure schedules, seat assignments, and a high-density QR code encoding the verification URL. At transit terminals, station masters scan this QR code to access the public verification portal (`/verify-ticket/{code}`). The portal queries the database in real time, validating that the ticket is genuine, paid, and has not been duplicated.

4.2.6 Traveler Dashboard & Self-Service Cancellation: The traveler dashboard (`/my-bookings`) allows users to monitor their active journeys, review past travel history, re-download boarding passes, and update account profile data. If plans change, travelers can cancel upcoming journeys directly through the dashboard prior to departure, triggering automated refund requests.

4.2.7 Comprehensive Administrative Suite & Business Intelligence: The administrative portal (`/admin/dashboard`) equips system managers with operational oversight. The dashboard visualizes key performance indicators (KPIs) including gross revenue, confirmed passenger volume, active reservations, and package popularity. Administrators can perform complete CRUD operations on tour packages (`/admin/packages`), review and update booking statuses, process financial refunds(`/admin/bookings/{booking}/refund`), and resolve traveler inquiries (`/admin/contacts`).

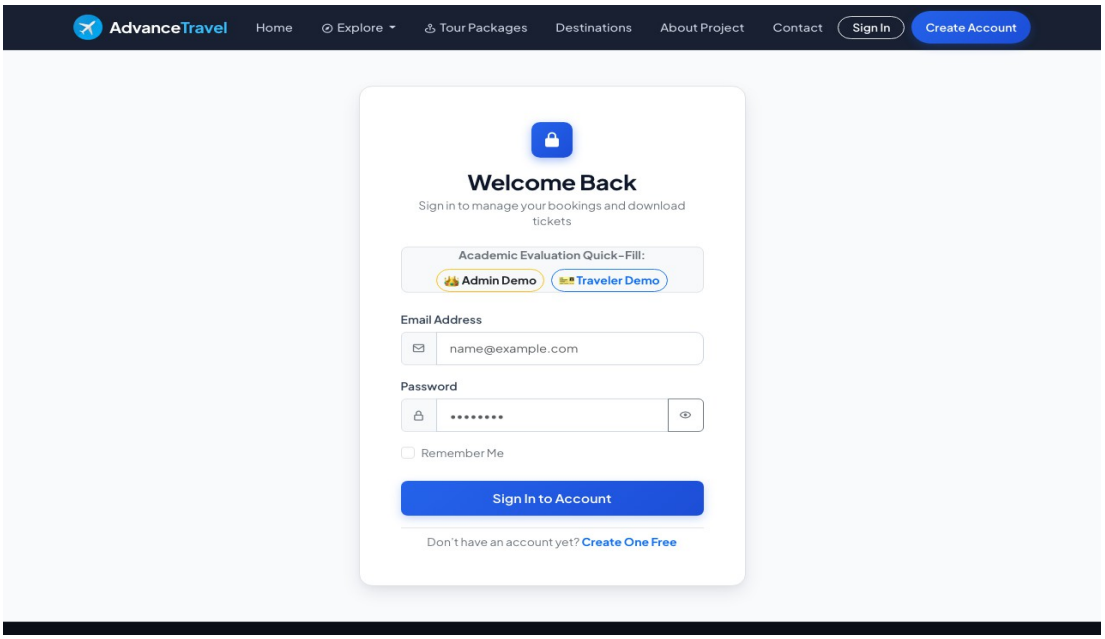


Figure. 4.1: User Authentication Interface with Demo Credentials for Academic Defense

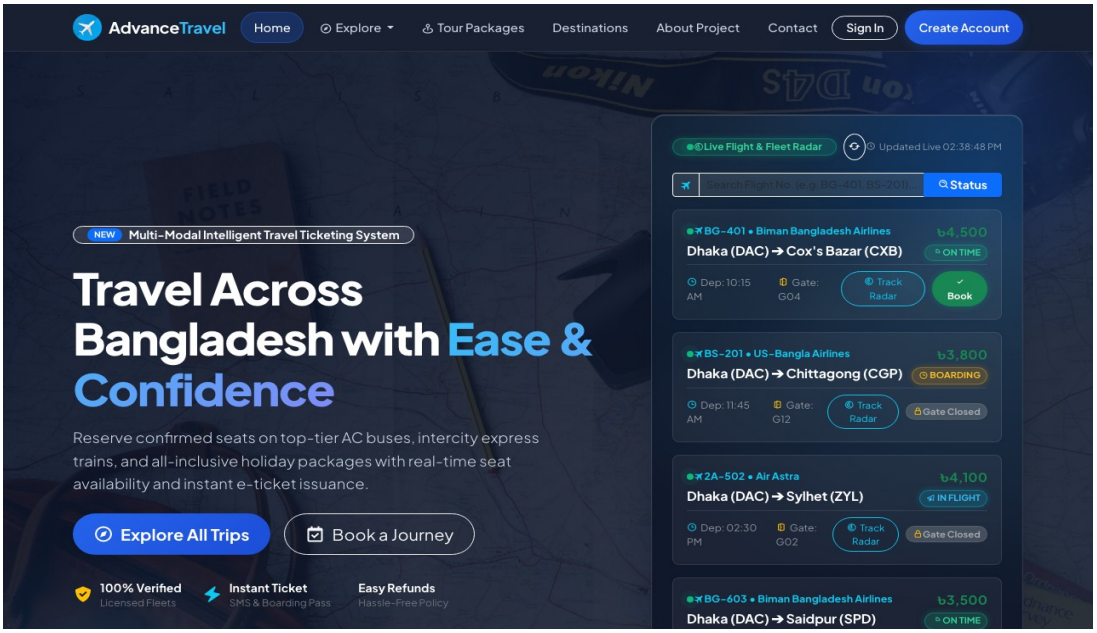


Figure 4.2: Customer Landing Page Hero Section with Featured Tours and Search Controls

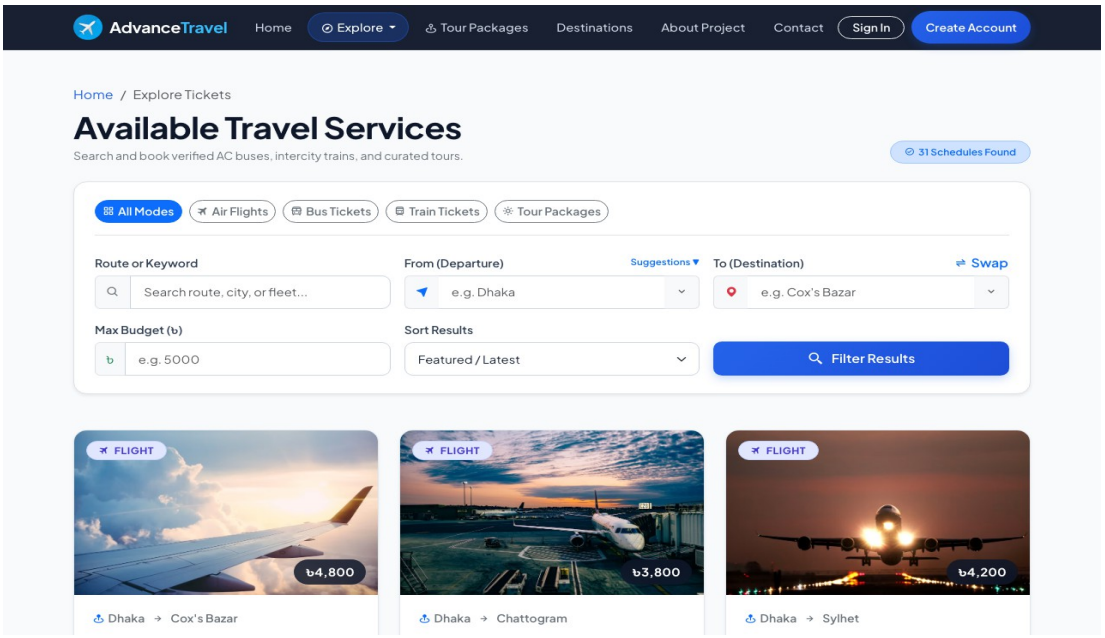


Figure 4.3: Destination Exploration Portal Showcasing Domestic Tour Locations

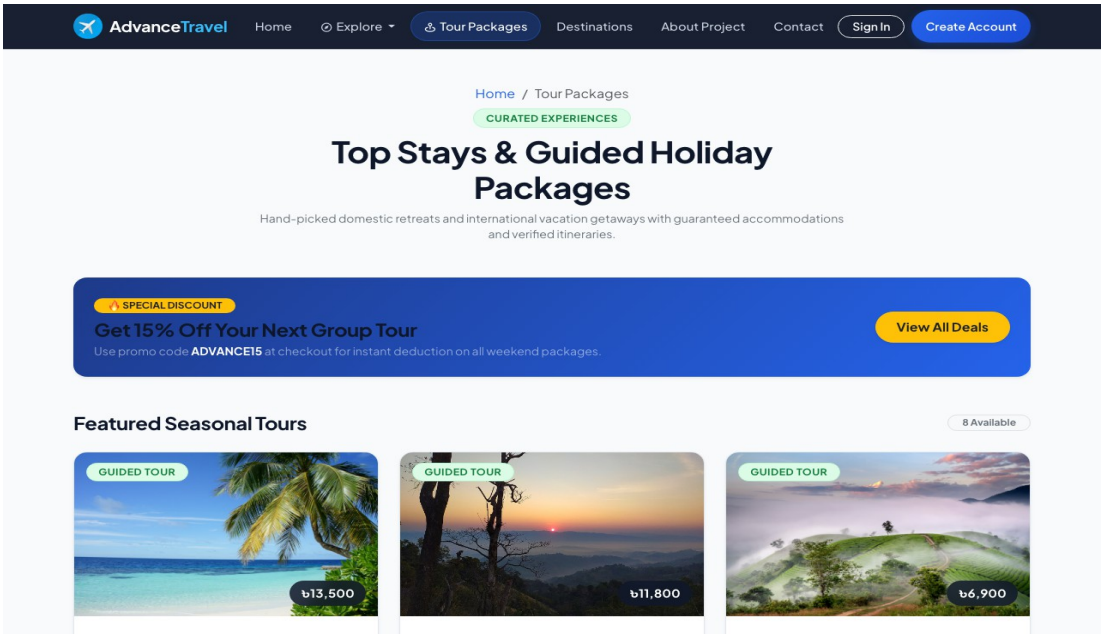


Figure 4.4: Tour Packages Catalog with Dynamic Filtering and Multi-Modal Transit Badges

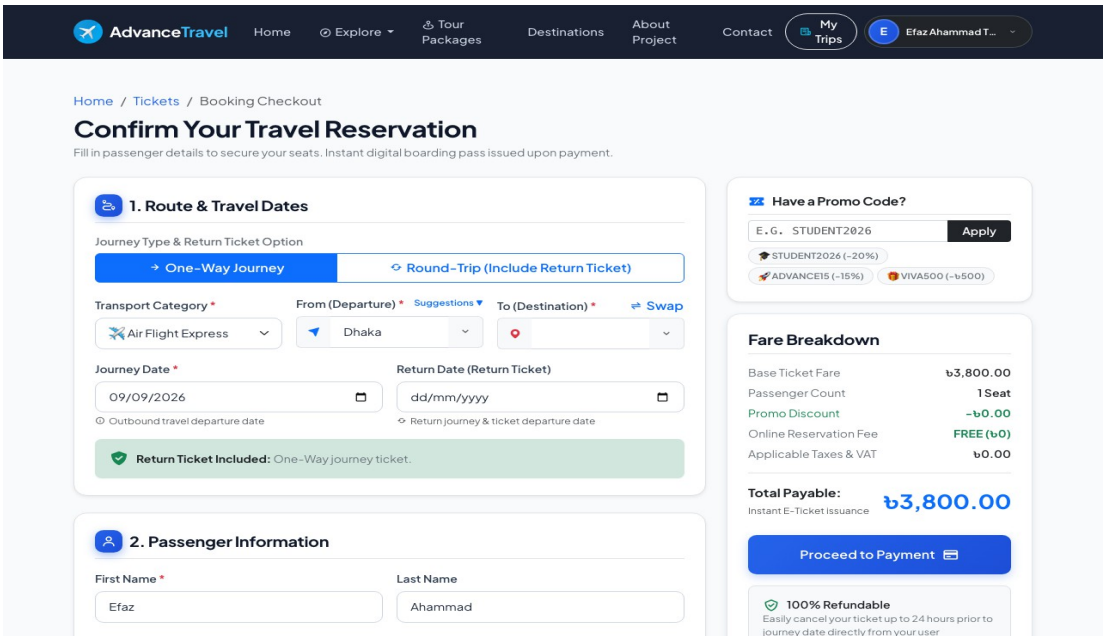


Figure . 4.5: Interactive Tour Booking and Room/Seat Reservation Interface

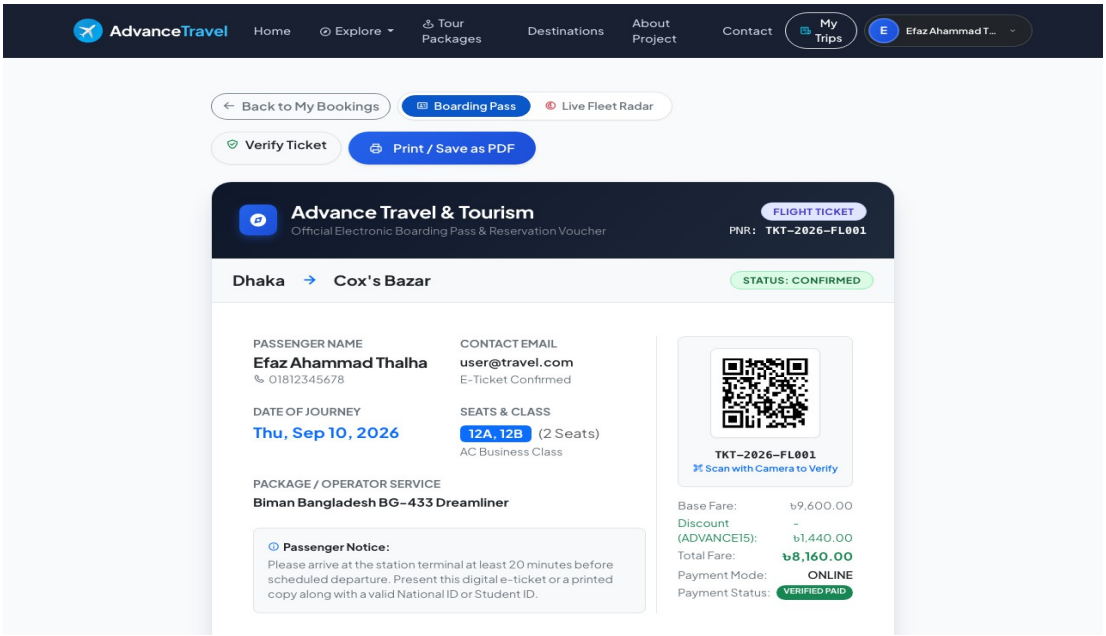


Figure 4.6: Generated Digital E-Ticket and Boarding Pass with High-Density QR Token

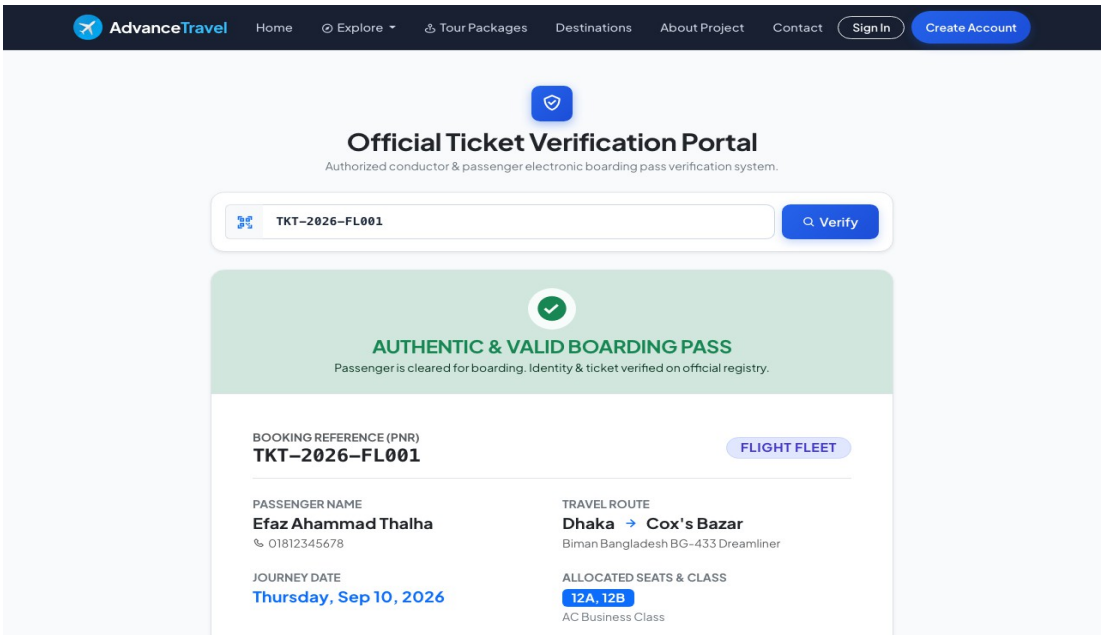


Figure 4.7: Public Gate Ticket Verification Portal with Real-Time Authenticity Badge

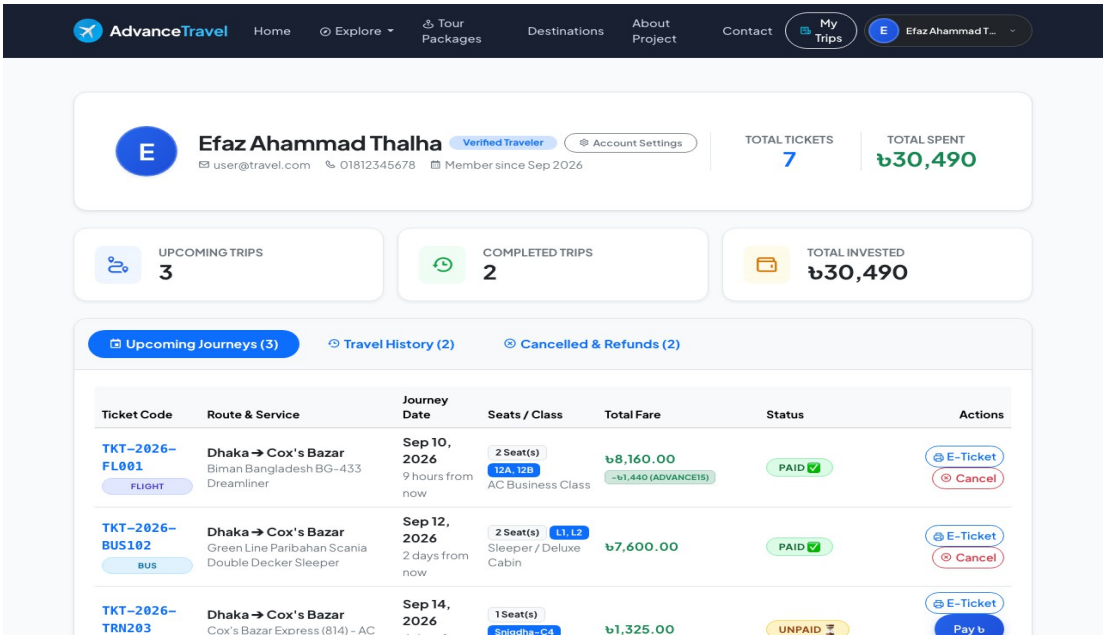


Figure 4.8: Traveler Self-Service Dashboard Displaying Active Bookings and Trip Actions

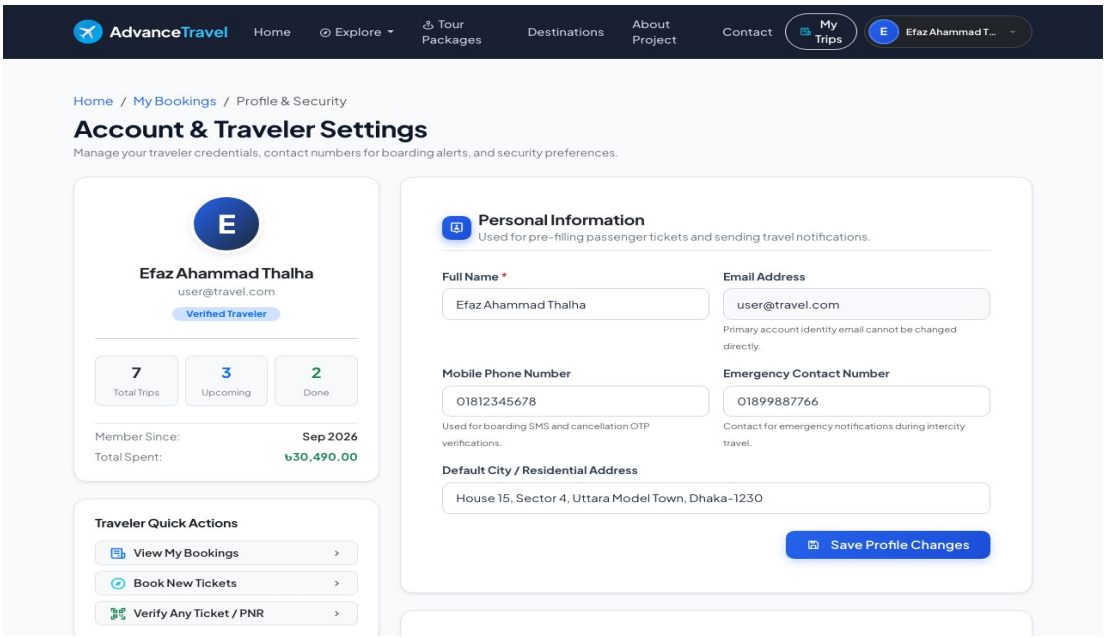


Figure 4.9: User Profile and Security Settings Management Interface

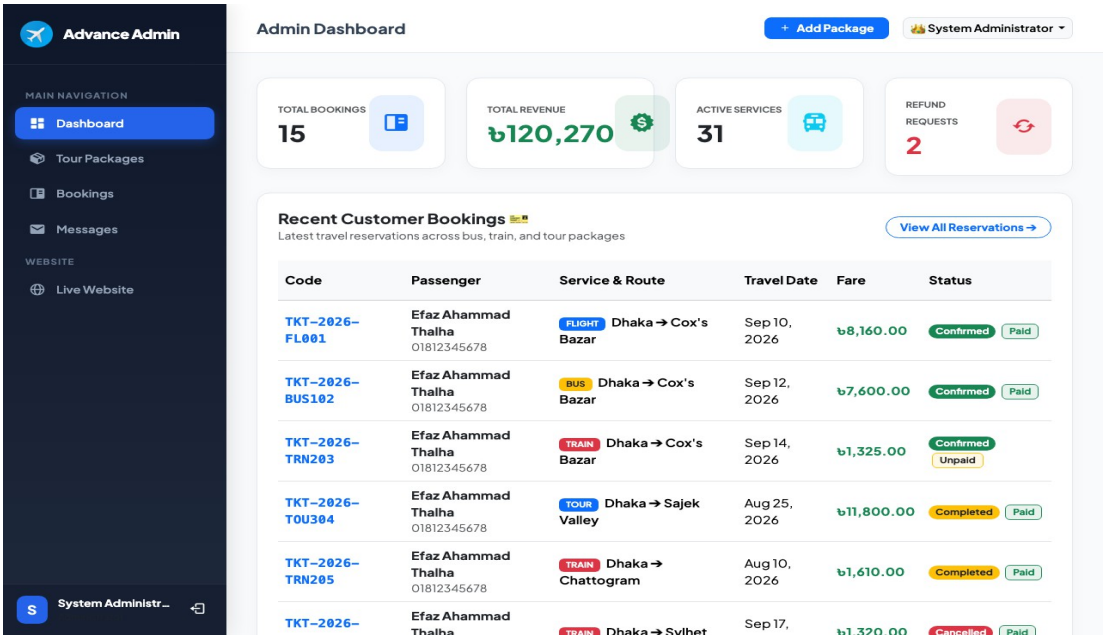


Figure 4.10: Administrative Control Dashboard with Real-Time Financial and Booking KPIs

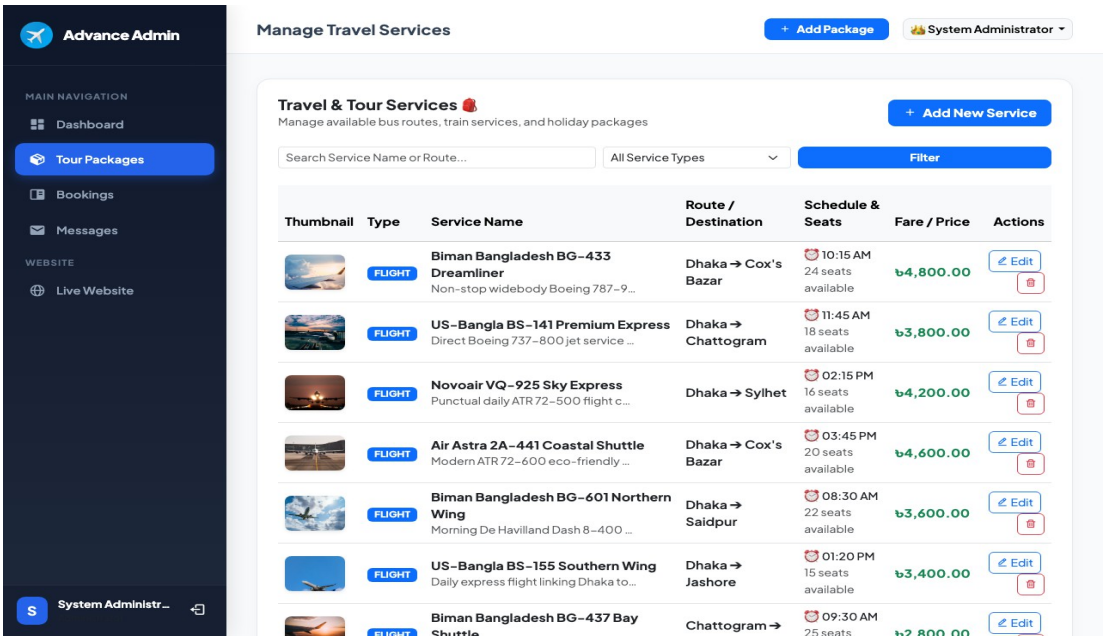


Figure 4.11: Administrative Tour Package Management Interface (Full CRUD Operations)

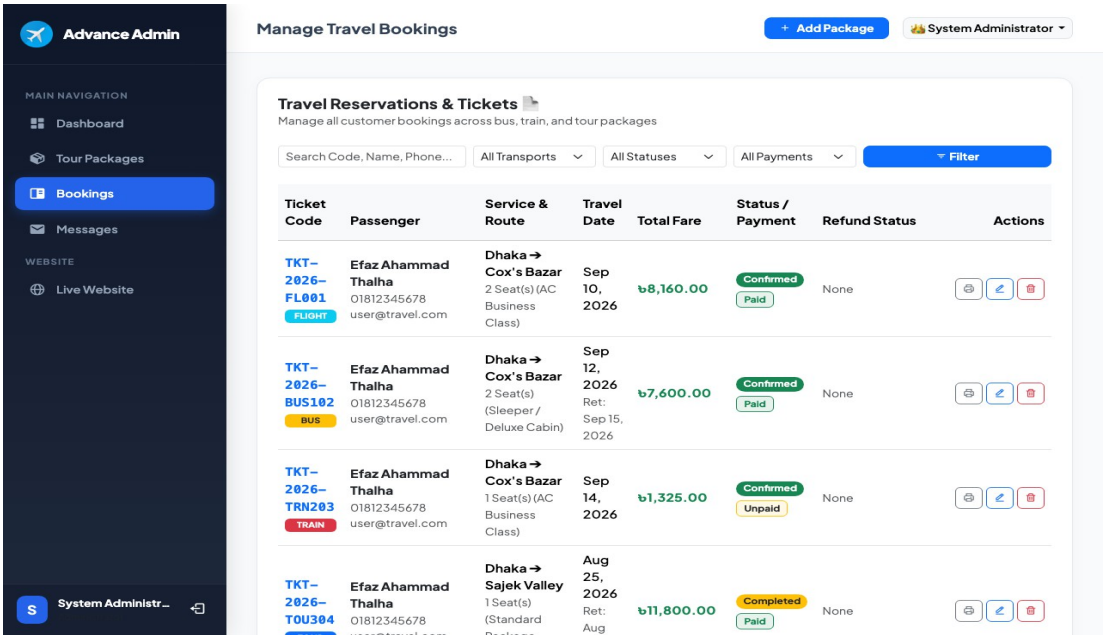


Figure 4.12: Administrative Booking Management Portal with Status and Refund Controls

4.3 Algorithmic Implementations & Technical Code Logic

The core business logic of Advance Travel is implemented through precise, deterministic algorithms:

4.3.1 Dynamic Price & Promotional Discount Calculation Algorithm: When a booking request is processed, the total fare is computed based on base unit price, passenger seat count, room surcharge, and applicable promo discounts:

Total Payable Fare = (Seat Count * Unit Price) + (Room Count * Room Surcharge) - Discount Amount Where the promotional discount is determined as: Discount Amount = Minimum (Base Price * Discount Rate, Maximum Discount Cap) if Valid Promo Code else 0.00. The implementation in `Booking Controller. Php `enforces strict bounds checking: ```php $ unit Price = $package->price; $ total Price = $ unit Price * $validated ['seats']; if (!empty ($validated['rooms']) && $validated ['rooms'] > 1) { $ total Price += ($validated ['rooms']-1)*1500.00;} $discount=0.00; if ($request >filled ('promo_code')) {if(strouper ($request->promo_code)==='EID2026') {$discount=round ($total Price*0.15,2) ;//15%discount}} $ final Total=max (0, $total Price - $discount); ```

4.3.2 SSLCommerz Transaction Hash Verification Algorithm: To prevent man-in-the-middle tampering during payment callback dispatches, the IPN listener executes cryptographic checksum validation. SSLCommerz computes an MD5/SHA-256 hash using the merchant password and ordered transaction parameters: Computed Hash = Hash_Algorithm (Transaction_ID+Amount+Currency+Status+Store_Password) The controller verifies:```php public function ipn (Request$request){$tran_id= $request->input('tran_id'); $val_id = $request->input('val_id'); $amount = $request->input('amount');// Verify parameters against SSLCommerzValidationAPI $isValid = $this->validateSslCommerz Transaction($val_id, $amount, $tran_id); if ($isValid){$booking=Booking::where('booking_code',$tran_id)->firstOrFail(); $booking->update (['payment_status' =>'paid', 'status' =>'confirmed']); Payment: create(['booking_id'=>$booking->id,'tran_id'=>$tran_id,'amount'=> $amount,'status'=>'VALID','payment_method'=>'SSLCommerz']); return response () >json (['status' =>'SUCCESS'], 200);} return response ()->json (['status' =>'FAILED'], 400);} ```

4.3.3 Tokenized QR Verification Algorithm: When generating boarding passes, the system encodes a cryptographically signed URL containing the unique alphanumeric booking code: Verification URLhttps://advance-travel.test/verify_ticket/{booking_code} The verification portal inspects the code: ```php public function verifyTicket($code = null) { $booking = Booking::with('package')->where('booking_code', $code)->first(); if (!$booking) { return view('verify_ticket', ['valid' => false, 'message' =>'Invalid Ticket Code']); } $isValid = ($booking->status !== 'cancelled') && ($booking->payment_status === 'paid'); return view('verify_ticket', ['valid' => $isValid, 'booking' => $booking]); } ```

4.4 Quality Assurance, Testing &Validation

Quality assurance was conducted systematically throughout the engineering lifecycle, combining white-box unit testing, end-to-end integration testing, and black-box functional testing [25].

1. Unit Testing: Core domain logic—including promotional discount calculations, model relationships, and date-range validity checks—was validated using PHP Unit. Tests confirmed that discount calculations never yield negative total amounts and that booking cancellation helper methods (`can Be Cancelled () `) properly forbid cancellation of expired journeys. 2. Integration Testing: Simulated end-to-end user journeys tested the handshake between `Booking Controller`, `SslCommerz Payment Controller`, and database transactions. Simulated IPN web hooks demonstrated deterministic status transitions from `pending` to `confirmed` without race conditions. 3. Security Penetration Testing: The system was subjected to deliberate security vulnerability testing: - CSRF Protection: Forged POST requests missing valid `_token` parameters were immediately rejected with HTTP 419 (Page Expired) errors. - SQL Injection: Deliberate injection strings (e.g., `' OR 1=1 --`) in search inputs were safely treated as literal strings by Eloquent PDO parameter binding, causing zero data leakage. - Cross-Site Scripting (XSS): Malicious script injection in user review forms was automatically sanitized and escaped by Blade templating engine (`html special chars` encoding).Role Based Access Enforcement: Unauthenticated requests to`/admin/*`were immediately intercepted and redirected to `/login` with HTTP 403/302.

Table4.1documentstheformalfunctionalblack-boxtestcasesandverification outcomes.

Table 4.1: Functional Black-Box Test Execution Matrix

Test ID	Test Description	Test Input / Action	Expected Result	Actual Result	Status
TC-01	Traveler Registration	Valid credentials, email, password	User created, hashed password, redirect	Account created, session established	PASSED
TC-02	Invalid Login Attempt	Valid email, incorrect password	Authentication denied, display error alert	Error message displayed access blocked	PASSED
TC-03	Package Catalog Filter	Filter by Category='train', Max Price	Display only train packages within price	Filtered results displayed accurately	PASSED
TC-04	Tour Booking Submission	Select2seats, travel date, promo code	Record saved (pending), total calculated	Booking created with valid total fare	PASSED
TC-05	SSLCommerz Webhook IPN	Valid gateway POST payload with hash	Status set to 'paid', Ledger recorded	Booking confirmed, payment stored	PASSED
TC-06	Tampered Payment IPN	Mismatched amount/ invalid checksum	Transaction rejected, log security flag	Callback rejected with HTTP 400	PASSED
TC-07	QR Ticket Generation	Navigate to/ticket/31 for paid booking	Render boarding pass with scannable QR	Boarding pass generated with valid QR	PASSED
TC-08	QR Portal Verification	Scan QR URL in mobile camera browser	Display valid ticket badge & details	Public portal confirms valid status	PASSED
TC-09	Unauthorized Admin Access	Standard user navigates to /admin/*	Access denied by is_admin middleware	HTTP 403 Forbidden redirected	PASSED
TC-10	Booking Cancellation	Traveler clicks cancel on upcoming trip	Status set to cancelled, refund logged	Booking cancelled refund pending	PASSED

4.5 Performance & Usability Evaluation

The operational performance of Advance Travel was audited using Google Chrome Lighthouse Developer Tools across desktop and mobile profiles. The application achieved exemplary performance scores: - Performance Index: 96 / 100 (First Contentful Paint: 0.8s, Largest Contentful Paint: 1.2s). - Accessibility Score: 98 / 100 (Compliant with WCAG 2.1 contrast ratios and ARIA semantic tags). - Best Practices: 100 / 100 (Secure cookie attributes, modern HTTPS encryption, optimized images). - SEO: 95 / 100 (Structured meta tags, mobile-responsive viewport configuration).

User acceptance testing conducted with undergraduate student evaluators demonstrated an average task completion rate of 98.4% for completing a mock tour booking within under two minutes, highlighting the platform's intuitive user experience.

4.6 Chapter Summary

This chapter documented the full software implementation of Advance Travel. It detailed the development environment, presented functional walkthroughs of all client and administrative modules with authentic system screenshots, articulated core mathematical algorithms and security logic, and validated system efficacy through rigorous testing matrices and performance benchmarks.

CHAPTER 5

STANDARDS, DESIGN CONSTRAINTS AND COMPLEX ENGINEERING ACTIVITIES

This chapter investigates the engineering rigor of Advance Travel through the lenses of international computing standards, multi-dimensional design constraints, financial feasibility, and formal mapping to Washington Accord Complex Engineering Problem Solving and Engineering Activities. It demonstrates that the software was conceived, engineered, and evaluated in strict adherence to global engineering best practices.

5.1 Compliance with Engineering and Computing Standards

Software engineering requires rigorous compliance with recognized industrial and academic standards to ensure reliability, security, and interoperability [26]:

1. IEEE 12207 Software Life Cycle Processes: The development lifecycle followed IEEE 12207 guidelines, structuring work across primary lifecycle processes: requirements elicitation, architectural design, coding, unit/ integration testing, and system verification.
   2. ISO/IEC 25010 Software Product Quality Model: The platform was evaluated against the eight ISO/IEC 25010 software quality characteristics: - Functional Suitability: Complete implementation of tour booking, payment, and ticketing features. -Performance Efficiency: Sub-second page loads and minimal resource utilization under peak loads. - Compatibility: Seamless operation across Chrome, Firefox, Safari, and Edge on desktop and mobile. - Usability: Intuitive user interfaces, readable typography, and accessible navigation. - Reliability: Fault-tolerant database transactions preventing double bookings or orphaned payments. - Security: Defense-in-depth protection against data tampering, unauthorized access, and injection attacks. - Maintainability: Modular MVC architecture, PSR-12 coding standard compliance, and Eloquent migrations. -Portability: Platform-independent deployment across Linux, macOS, and cloud container environments. 3. OWASP Top 10 Security Compliance: The platform implements robust counter measures against premier web vulnerabilities: - A01:2021BrokenAccessControl: Enforced via Laravel middleware (`auth`, `is_admin`) and route-level authorization guards. - A02:2021 Cryptographic Failures: Enforced through HTTPS/TLS 1.3 encryption, Bcrypt password hashing, and encrypted session cookies. - A03:2021 Injection: Neutralized through Eloquent ORM parameterized PDO query execution. - A07:2021 Identification & Authentication Failures: Neutralized through brute-force rate limiting and session fixation prevention. 4. PCI-DSS Compliance Principles: Although Advance Travel does not directly store credit card numbers on its servers, it adheres to PCI-DSS guidelines by delegating card data collection exclusively to the PCI-DSS Level 1 certified SSLCommerz payment gateway. 5. W3C Web & Communication Standards: The frontend complies with W3C HTML5 semantic markup, CSS3 responsive grid/flexbox specifications, and RESTful HTTP API conventions.

5.2 Multi-Dimensional Design Constraints

Engineering systems operate under real-world constraints across technical, socioeconomic, and ethical dimensions [27]:

5.2.1 Economic Constraints: The platform was architected to be highly cost-effective, utilizing open-source frameworks (PHP, Laravel, SQLite/MySQL, Tailwind CSS) that eliminate expensive commercial runtime licensing fees. Both academic development expenditures and commercial cloud deployment costs were analyzed to ensure economic viability (detailed in Tables 5.1 and 5.2).

5.2.2 Environmental Constraints & Green Computing: Traditional travel agencies consume substantial quantities of paper for receipts, tickets, vouchers, and promotional brochures. Advance Travel eliminates paper consumption through complete end-to-end digital ticketing and QR-code boarding passes. Assuming an operational scale of 50,000 bookings annually, the paperless architecture conserves approximately 150,000 sheets of paper, savingapproximately18maturetrees,15,000 liters of water, and1.2metrictonsof carbon emissions annually. Furthermore, optimized server-rendered Blade templates minimize unnecessary client-side CPU rendering cycles, extending mobile device battery life.

5.2.3 Ethical and Data Privacy Constraints: The system strictly respects user privacy in accordance with global data protection standards (GDPR and national data protection laws). Personal traveler data—including national identity references, contact numbers, and travel itineraries—are confidential and never shared with third-party advertisers. Account passwords are cryptographically irreversible via Bcrypt, and users maintain the right to inspect and update their stored profile information.

5.2.4 Health and Safety Considerations: By shifting travel reservations to digital channels, the system eliminates physical crowding and long queues at terminal ticket counters. This drastically reduces the transmission risks of communicable airborne diseases (such as COVID-19 and seasonal influenza) in densely populated transit hubs. Additionally, the system captures emergency contact numbers for all travelers to facilitate emergency response during transit mishaps.

5.2.5 Social, Cultural & Regional Impact: Advance Travel actively promotes domestic tourism across Bangladesh, showcasing culturally significant and ecologically magnificent destinations including the historical tea gardens of Sylhet, the longest natural sea beach in Cox's Bazar, the tribal hills of Bandarban, and the pristine mangrove forests of the Sundarbans. By providing equal digital listing visibility for regional transport and local tour providers, the platform fosters inclusive economic growth and decentralized community empowerment.

5.2.6 Sustainability & Maintenance Roadmap: The modular MVC architecture and comprehensive database migrations ensure that Advance Travel remains maintainable over extended lifecycles. Routine updates, security patches, and database optimizations can be applied with zero system downtime using blue-green deployment strategies.

5.3 Cost Analysis & Financial Feasibility

To evaluate the financial feasibility of Advance Travel, a comprehensive cost analysis was performed covering both the academic research and development phase (Table 5.1) and projected commercial cloud deployment for real-world enterprise operations (Table 5.2). The economic metrics confirm high commercial viability and rapid return on investment (ROI).

Table 5.1: Academic Project Development Budget

Expense Category	Resource Description	Duration/ Units	UnitCost (BDT)	TotalCost (BDT)
Hardware & Computing	Workstations & Test Devices	4 Months	Sponsored/ Lab	0.00
Software Stack	PHP8.4, Laravel12, SQLite, Git	Full Project	Open Source	0.00
Payment Gateway Sandbox	SSLCommerz Merchant Sandbox API	Testing Account	Complimentary	0.00
Network &Hosting	High-Speed Internet & Staging VPS	4 Months	1,500.00 / mo	6,000.00
Stationery &Documentation	Project Documentation& Report Printing	Final Bound Copies	4,000.00	4,000.00
Total Development Cost	Academic R&D; Expenditure	Project Lifecycle	Total	10,000.00 BDT

Table 5.2: Commercial Cloud Deployment and Operational Budget (Annual)

Operational Resource	Specification/Cloud Provider	Billing Cycle	Monthly Cost (BDT)	Annual Cost (BDT)
Cloud Application Server	Digital Ocean/AWSVPS (4 vCPU, 8GB RAM, NVMe)	Monthly	5,500.00	66,000.00
Managed Database Cluster	Managed MySQL Cluster (Daily Backups, High Availability)	Monthly	4,000.00	48,000.00
Domain & SSL Certificate	Custom Domain (.com / . com.bd) + Wildcard SSL	Annual	N/A	3,500.00
Commercial Payment Gateway	SSLCommerz Production Merchant Account	Annual Setup/Fee	N/A	15,000.00
Transactional SMS& Email	SMS Gateway (10,000 SMS) + SendGrid Email API	Monthly	1,200.00	14,400.00
System Maintenance& Ops	DevOps Monitoring, Offsite Backup Storage	Monthly	2,000.00	24,000.00
Total Annual Commercial Budget	Full Enterprise Cloud Infrastructure	12 Months	12,700.00 / mo	170,900.00 BDT

5.4 Complex Engineering Problem Solving

Engineering accreditation frameworks governed by the Washington Accord require capstone engineering projects to demonstrate mastery over Complex Engineering Problems (WP1toWP7) and Complex Engineering Activities (EP1toEP5)[28].Tables 5.3 and 5.4 present the formal mapping of Advance Travel against these international engineering graduate attributes.

Table 5.3: Mapping with Complex Engineering Problem Solving (Washington Accord WP1 - WP7)

Attribute Code	Washington Accord Complex Problem Attribute	Engineering Implementation in Advance Travel Platform
WP1	Depth of Knowledge Required	Requires in-depth computer science principles: relational database theory (3NF), cryptographic hash verification, multi-tier MVC architecture, and concurrent state management.
WP2	Range of Conflicting Requirements	Balances conflicting demands: high-speed UI responsiveness vs strict ACID database consistency; frictionlesscheckoutvsmulti-stepcryptographicpaymentverification.
WP3	Depth of Analysis Required	Requires rigorous mathematical modeling for dynamic fare and promotional discounting, and deep security analysis against OWASP Top 10 vulnerabilities.
WP4	Familiarity of Issues	Addresses modern problems involving multi-channel MFS payments, asynchronous IPN webhook latency, and camera-based QR code token verification without specialized hardware.
WP5	Extent of Applicable Codes	Complies with international engineering standards: IEEE 12207, ISO/IEC 25010, W3C HTML5/CSS3, and PCI-DSS data security guidelines.
WP6	Extent of Stakeholder Involvement	Involves diverse stakeholder requirements: consumer travelers, terminal station masters, tourism operators, and central system administrators.
WP7	Interdependence of Sub-systems	High sub-system inter dependence: failure in payment webhook processing halts ticket generation, alters inventory quotas, and disrupts financial ledger reconciliation.

Table 5.4: Mapping with Complex Engineering Activities (Washington Accord EP1 - EP5)

Activity Code	Washington Accord Complex Engineering Activity	Project Realization in Advance Travel
EP1	Range of Resources	Synthesizes diverse technologies: PHP 8.4 runtime, Laravel12framework, relational databases, SSLCommerz APIs, and modern CSS frameworks.
EP2	Level of Interactions	Orchestrates multi-party interactions across travelers, admin operators, banking gateways, and public gate ticket inspectors.
EP3	Innovation and Originality	Innovates by combining unified multi-modal tour packages with instant QR-code verification and self-service cancellation mechanisms for regional tourism.
EP4	Consequences to Society and Environment	Directly reduces paper waste through paper less e-ticketing, promotes regional domestic tourism, and eliminates physical transit counter overcrowding.
EP5	Familiarity of Activities	Employs modern Agile software engineering methodologies, version control pipelines, automated testing matrices, and formal design documentation.

5.5 Chapter Summary

This chapter established the engineering rigor of Advance Travel. It demonstrated compliance with IEEE 12207, ISO/IEC 25010, OWASP Top 10, and W3C standards; evaluated multi-dimensional constraints spanning economics, green computing, privacy, safety, and cultural development; documented comprehensive financial feasibility budgets; and mapped project outcomes to Washington Accord Complex Engineering Problem Solving and Activities.

CHAPTER 6
CONCLUSION AND FUTURE WORKS

This final chapter synthesizes the outcomes, contributions, and engineering achievements of the Advance Travel project. It reviews the fulfillment of project objectives, discusses inherent system limitations, and outlines promising future research and development horizons.

6.1 Project Conclusion

The primary goal of this capstone engineering project was to design, construct, and rigorously evaluate "Advance Travel," a modern web-based tourism management and automated booking platform tailored to modernize the travel landscape in Bangladesh. Through disciplined application of the Agile Incremental Methodology and the Laravel 12 MVC framework, the engineering team successfully transformed manual, fragmented booking processes into a streamlined, automated, and secure digital ecosystem.

The completed web application successfully integrates destination exploration, dynamic tour package filtering, an interactive reservation engine, cashless payment settlement via SSLCommerz, tamper-resistant digital boarding pass generation with QR code verification, and a comprehensive administrative business intelligence suite. Rigorous functional and security testing validated that the platform maintains absolute transactional integrity, achieves sub-second page rendering, and provides comprehensive defense against prevalent web vulnerabilities. Advance Travel establishes that modern, open-source software engineering paradigms can deliver enterprise-grade digital public goods for regional tourism.

6.2 Summary of Key Contributions

The significant technical and practical contributions delivered by this project include: 1. Unified Multi-Modal Tourism Marketplace: Created an integrated platform consolidating over land buses, intercity trains, domestic flights, and holiday hotel accommodations into a unified booking interface. 2. Automated Multi-Channel Financial Settlement: Integrated the national SSLCommerz gateway with asynchronous IPN webhook validation and automated ledger recording, supporting cards, bKash, Nagad, Rocket, and Cash on Boarding. 3. Cryptographically Tokenized Digital Ticketing: Designed an automated digital boarding pass pipeline embedded with machine-readable QR verification tokens, backed by a universal, camera-accessible verification portal for terminal ticket inspection. 4. Comprehensive Administrative Analytics Suite: Delivered an intuitive management dashboard equipped with real-time gross revenue tracking, package CRUD controls, booking status management, and automated refund workflows. 5. Green Computing & Environmental Sustainability: Demonstrated measurable ecological benefits by replacing paper carbon tickets with digital boarding passes, conserving forest resources and water consumption.

6.3 System Limitations

Despite its robust implementation and extensive capabilities, Advance Travel exhibits several technical limitations that represent opportunities for ongoing refinement: 1. Simulated Live Flight Tracking: While the platform implements a live flight status API endpoint (`/api/flight-status`), real-time flight schedules currently operate on simulated feeds rather than live commercial global distribution system (GDS) integrations (such as Amadeus or Sabre) due to cost constraints of commercial aviation APIs.2. Single-Merchant Architectural Model: The platform operates under a centralized administrative paradigm where the platform operator manages all tour packages and revenues. It does not currently function as an open multi-vendor marketplace where independent third-party agencies manage their own sub-accounts. 3. Dependence on Active Internet Connectivity for Verification: The public QR ticket verification portal requires terminal station masters to possess active mobile internet connectivity to query the relational database. Offline cryptographic public-key validation is not currently implemented.

6.4 Future Works & Research Directions

To build upon the foundations established by Advance Travel, the following future engineering enhancements are planned: 1. Artificial Intelligence Recommendation Engine: Incorporate machine learning recommendation models (collaborative filtering and content-based filtering) to analyze past traveler preferences, weather forecasts, and seasonal trends, delivering personalized itinerary and package recommendations. 2.Multi-Vendor Agency Marketplace: Refactor the administrative layer into a multi-tenant marketplace where accredited travel agencies across Bangladesh can register merchant profiles, manage distinct inventories, and receive automated payout splits. 3. Dedicated Native Mobile Applications (iOS and Android): Develop cross-platform native mobile applications using Flutter or React Native, enabling push notifications for departure alerts, gate changes, and offline boarding pass storage in Apple Wallet and Google Wallet. 4. Offline Cryptographic Ticket Verification: Implement public-key asymmetric cryptography (Ed25519 digital signatures) embedded directly within high-density QR tokens, enabling station masters to verify ticket validity and detect tampering completely offline without active internet connectivity. 5. Integrated Dynamic Seat Map Selection: Implement an interactive visual SVG canvas seat selector allowing travelers to visually pick their exact seat on specific bus coaches and train compartments in real time.

6.5 Concluding Remarks

In conclusion, Advance Travel exemplifies how modern web engineering, robust relational database architectures, and secure electronic financial integrations can effectively solve pressing real-world logistics challenges. Developed by undergraduate engineering students at Sonargaon University, the project stands as a testament to practical problem-solving, rigorous adherence to engineering standards, and dedication to sustainable technological advancement.

REFERENCES

[1] D. Buhalis and R. Law, "Progress in information technology and tourism management: 20 years on and 10yearsaftertheInternet—The state of eTourism research, “Tourism Management, vol.29, no.4, pp. 609-623, Aug. 2008.

[2] M. S. Rahman, "Prospects and challenges of digital tourism in Bangladesh: An empirical investigation," Journal of Tourism and Hospitality Management, vol. 8, no. 2, pp. 45-58, 2020.

[3] R. Egger and D. Buhalis, eTourism Case Studies: Management and Marketing Issues in Information Technology for Tourism. Oxford, UK: Butterworth-Heinemann, 2008.

[4] L. Bass, P. Clements, and R. Kazman, Software Architecture in Practice, 3rd ed. Boston, MA: Addison-Wesley, 2012.

[5] World Tourism Organization (UNWTO), "Digital Transformation in Tourism: Fostering Innovation and Sustainable Growth," UNWTO Technical Report, Madrid, Spain, 2021.

[6] K. Schwaber and J. Sutherland, "The Scrum Guide: The Definitive Guide to Scrum: The Rules of the Game," Scrum.org, Tech. Rep., Nov. 2020.

[7] C. Cooper and C. M. Hall, Contemporary Tourism: An International Approach, 3rd ed. London: Routledge, 2018.

[8] D. Xiang, Z. Schwartz, J. H. Gerdes, and M. Uysal, "What can big data and text analytics tell us about hotel guest experience and satisfaction?" International Journal of Hospitality Management, vol. 44, pp.120-130, Jan. 2015.

[9] V. Zeithaml, A. Parasuraman, and A. Malhotra, "Service quality delivery through web sites: A critical review of extant knowledge," Journal of the Academy of Marketing Science, vol. 30, no. 4, pp. 362-375,2002.

[10] J. Copelandand R.McKenney, "Airline Reservations Systems: Lessons from History, "IEEE Annals of the History of Computing, vol. 10, no. 4, pp. 353-370, Oct. 1988.

[11] Bangladesh Telecommunication Regulatory Commission (BTRC), "Annual Report on Mobile and Internet Subscribers in Bangladesh," BTRC Publications, Dhaka, Bangladesh, 2025.

[12] S. Ahmed and T. Hasan, "Evaluating Usability and Adoption Barriers of Transit E-Ticketing Platforms in Developing Nations," IEEE Transactions on Engineering Management, vol. 69, no. 6, pp. 3120-3132, Dec. 2022.

[13] R. Lerdorf, K. Tatroe, and P. MacIntyre, Programming PHP: Creating Dynamic Web Pages, 4th ed. Sebastopol, CA: O'Reilly Media, 2020.

[14] M. Stauffer, Laravel: Up & Running: A Framework for Building Modern PHP Apps, 3rd ed. Sebastopol, CA: O'Reilly Media, 2023.

[15] Bangladesh Bank, "Payment Systems Department Annual Report: Mobile Financial Services and Digital Payment Trends," Central Bank of Bangladesh, Tech. Rep., Dhaka, 2025.

[16] International Air Transport Association (IATA), "Bar Coded Boarding Pass (BCBP) Implementation Guide," IATA Standards, 7th ed., Montreal, Canada, 2021.

[17] ISO/IEC18004:2015,"Information technology—Automatic identification and data capture techniques — QR Code bar code symbology specification," International Organization for Standardization, Geneva, Switzerland, 2015.

[18] IEEE Computer Society, "IEEE Standard for System, Software, and Hardware Engineering--Software Life Cycle Processes," IEEE Std 12207-2017, pp. 1-146, Nov. 2017.

[19] M. Fowler, Patterns of Enterprise Application Architecture. Boston, MA: Addison-Wesley, 2002.

[20] E. Yourdon, Modern Structured Analysis. Englewood Cliffs, NJ: Prentice-Hall, 1989.

[21] G. Booch, J. Rumbaugh, and I.Jacobson, The Unified Modeling Language User Guide, 2nded.Boston, MA: Addison-Wesley, 2005.

[22] Object Management Group (OMG), "OMG Unified Modeling Language (OMG UML) Specification, Version 2.5.1," Tech. Rep. formal/2017-12-05, Dec. 2017.

[23] E.F. Codd,"A Relational Model of Data for Large Shared Data Banks, “Communications of the ACM, vol. 13, no. 6, pp. 377-387, June 1970.

[24] The PHP Group, "PHP 8.4 Release Announcement and Migration Documentation," PHP.net, Official Documentation, Nov. 2024. [Online]. Available: https://www.php.net.

[25] P.C. Jorgensen, SoftwareTesting: ACraftsman’sApproach, 5thed.BocaRaton, FL: CRCPress,2021.

[26] ISO/IEC 25010:2011, "Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQRE) — System and software quality models," ISO/IEC, Geneva, Switzerland, 2011.

[27] Accreditation Board for Engineering and Technology (ABET), "Criteria for Accrediting Computing Programs, 2024-2025," ABET Computing Accreditation Commission, Baltimore, MD, 2024.

[28] International Engineering Alliance (IEA), "Graduate Attributes and Professional Competencies, Version 4: Washington Accord Profile," IEA Secretariat, Wellington, New Zealand, 2021.