PRESIDENCY UNIVERSITY

Private University Estd. In Karnataka State by Act No. 41 of 2013
Itgalpura, Rajankunte, Yelahanka, Bengaluru – 560064

ONE STOP SOLUTION FOCUSING ON TOURISM

A PROJECT REPORT

Submitted by

KARTHIK	20231CSE3047
HARSHA S K	20231CSE3021
H N HEMANTH	20231CSE3025

Under the guidance of,

Mr. Asad Mohammed Khan

Assistant Professor

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

PRESIDENCY UNIVERSITY
BENGALURU
DECEMBER 2025

PRESIDENCY UNIVERSITY

Private University Estd. in Karnataka State by Act No. 41 of 2013
Itgalpura, Rajankunte, Yelahanka, Bengaluru – 560064

PRESIDENCY SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

Certified that this report **"One Stop Solution Focusing on Tourism"** is a bonafide work of **"KARTHIK (20231CSE3047), HARSHA S K (20231CSE3021), H N HEMANTH (20231CSE3025)"**, who have successfully carried out the project work and submitted the report for partial fulfilment of the requirements for the award of the degree of **BACHELOR OF TECHNOLOGY** in **COMPUTER SCIENCE AND ENGINEERING** during **2025-26.**

Asad m. Khan

Mr. Asad Mohammed Khan
Project Guide
Presidency School of Computer Science and Engineering Presidency University

4/05/25

Mr. Muthuraju V
Program Project Coordinator
PSCS
Presidency University

~~**Dr. Sampath A K**~~
Dr. Geetha A
School Project Coordinators
PSCS
Presidency University

Dr. Blessed Prince
Head of the Department
PSCS
Presidency University

Dr. Shakkeera L
Associate Dean
PSCS
Presidency University

Dr. Duraipandian N
Dean
PSCS & PSIS
Presidency University

Examiners

Sl. no.	Name	Signature
1	**Dr. Praveena K N**	Praveena K.N
2	**Dr. Sreelatha**	

PRESIDENCY UNIVERSITY

PRESIDENCY SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

DECLARATION

We the students of final year B. Tech in COMPUTER SCIENCE AND ENGINEERING at Presidency University, Bengaluru, named KARTHIK (20231CSE3047), HARSHA S K, (20231CSE3021) H N HEMANTH (20231CSE3025), hereby declare that the project work titled **"One Stop Solution Focusing on Tourism"** has been independently carried out by us and submitted in partial fulfillment for the award of the degree of B. Tech in COMPUTER SCIENCE ENGINEERING during the academic year of 2025-26. Further, the matter embodied in the project has not been submitted previously by anybody for the award of any Degree or Diploma to any other institution.

NAME	USN	SIGNATURE
KARTHIK	20231CSE3047	Karthik
HARSHA S K	20231CSE3021	Harsha
H N HEMANTH	20231CSE3025	

PLACE: BENGALURU

DATE: 04-December 2025

ACKNOWLEDGEMENT

For completing this project work, We/I have received the support and the guidance from many people whom I would like to mention with deep sense of gratitude and indebtedness. We extend our gratitude to our beloved **Chancellor, Pro-Vice Chancellor, and Registrar** for their support and encouragement in completion of the project.

I would like to sincerely thank my internal guide **Mr. Asad Mohammed Khan, Assistant Professor**, Presidency School of Computer Science and Engineering, Presidency University, for his moral support, motivation, timely guidance and encouragement provided to us during the period of our project work.

I am also thankful to **Dr. Blessed Prince, Head of the Department, Presidency School of Computer Science and Engineering** Presidency University, for his mentorship and encouragement.

We express our cordial thanks to **Dr. Duraipandian N**, Dean PSCS & PSIS, **Dr. ShakkeeraL**, Associate Dean, Presidency School of computer Science and Engineering and the Management of Presidency University for providing the required facilities and intellectually stimulating environment that aided in the completion of my project work.

We are grateful to **Dr. Sampath A K, and Dr. Geetha A, PSCS** Project Coordinators, **Mr. Muthuraju V, Program Project Coordinator**, Presidency School of Computer Science and Engineering, or facilitating problem statements, coordinating reviews, monitoring progress, and providing their valuable support and guidance.

We are also grateful to Teaching and Non-Teaching staff of Presidency School of Computer Science and Engineering and also staff from other departments who have extended their valuable help and cooperation

KARTHIK
HARSHA S K
H N HEMANTH

ABSTRACT

Tourism can bring economic benefits to a region, promote cultural exchange, and create a better experience for the traveler. Unfortunately, the current systems for planning and managing a trip are often fragmented, with little or no trust associated with the information provided and many inefficiencies in the process. This paper presents a new integrated approach to the Tourism platform that aims to streamline and improve the entire travel experience, from trip planning to travel and enjoy destination.

This new platform will provide all the features needed to create an itinerary, receive real-time updates during your trip, and book a flight, hotel, and local activities, all using one platform. Working together with locals and promoting an environmentally sustainable product will provide the traveler with an opportunity to experience a more culturally enriched trip while helping to conserve the local ecosystem.

In addition, this platform is designed for ease of use, while placing emphasis on user security as well as creating opportunities for all users. Because of this, it will make it possible for users with varying cultural and personal preferences to participate equally in the trip planning process. The resulting product will provide users with greater satisfaction, reduce time spent planning a trip, and ultimately assist in promoting the development of sustainable tourism practices.

TABLE OF CONTENT

LIST OF FIGURES

LIST OF TABLES

ABBREVIATIONS

CIS	Center For Internet Security
NIST	National Institute of Standards and Technology
GUI	Graphical user interface
JSON	JavaScript Object Notation
API	Application Programming Interface
CSV	Comma Separated Values
CI/CD	Continuous Integration and Continuous Delivery/Deployment
SCAP	Security Content Automation Protocol
XML	Extensible Markup Language
OS	Operating System
OSCAT	One Stop CIS-Benchmark Auditing Tool
CPU	Central Processing Units
RAM	Random Access Memory
CLI	Command Line Interface
ISO	International Organization for Standardization
POSIX	Portable Operating System Interface
IoT	Internet of Things
I/O	Input/Output
VM	Virtual Machine
LTS	Long-term support
VS	Visual Studio
SSH	Secure Socket Shell
DISA	Defense Information Systems Agency
GDPR	General Data Protection Regulation
DPDPA	India's Digital Personal Data Protection Act
MIT	Massachusetts Institute of Technology
ATIG	Security Technical Implementation Guide
PDF	Portable Document Format
UFW	uncomplicated firewall

CHAPTER 1
INTRODUCTION

In today's world people travel a lot to different cities and countries. But it is hard for them to travel in the city without prior knowledge. This is the prior motive behind making this app. A lot of websites and apps exist that do the similar job. An app exists that can book hotels, another for booking cabs, another for booking shows, events etc but what we want to do is to integrate all of this into one app in order to save time of the user. Similar apps do exist but they are not very efficient and are very time consuming. We aim to build an app that is user friendly and easy to use and will help people find out what they are looking for in an easier and time efficient way.

1.1 Background

The importance of tourism in the overall economies and cultures of the world is an ever-present theme in business today. As digitalization continues to grow, the application of digital technologies in the travel and tourism industry is becoming increasingly common, including the use of mobile applications and AI recommendations, the integration of IoT devices into travel management and logistics, as well as the implementation of Smart Tourism Platforms. According to the World Travel and Tourism Council (WTTC, 2023), tourism accounts for more than 10% of the world economy (i.e., global GDP) and supports roughly one in ten jobs worldwide. The integration of digital tools helps travellers plan customized trips, find sustainable destinations, and improve overall travel experiences.

1.2 Statistics

In the year 2020, tourism generated approximately $41 billion in revenue for the country of India (source: Ministry of Tourism 2023), accounting for approximately 5.8% of total jobs and 6.8% of Gross Domestic Product (GDP). In the state of Karnataka alone, over 200 million domestic tourists visited in 2022, which provides an opportunity to develop additional digital infrastructure and allow for the creation of “one-stop” travel management websites. While the tourism industry has seen strong growth, several issues still exist; most notably, tourism information is generally disbursed among several platforms, information about tourism in rural areas is often limited, and there are few to no systems in place to provide real-time recommendations to those who may wish to visit.

1.3 Prior Existing Technologies

There are many services currently available that allow travelers to book hotel rooms and leave reviews on their favorite places (e.g., MakeMyTrip, TripAdvisor, Google Travel), but the current services are generally only used to create itineraries, book hotels, and find activities, and do not include community-integrated, locally-available, real-time recommendations, environmental sustainability, or itinerary advisement/management systems. As technology continues to improve, there are now many new types of technology available to enhance the traveler's experience, including the use of artificial intelligence (AI) in chatbots, geographic information systems (GIS) to create routings, and augmented reality (AR) guides.

1.4 Proposed Approach

Project Aim:

- To develop and also launch a Smart Tourism Platform that recognizes the need for integrating trip planning, hotel booking, local travel recommendations, environmental sustainability, and also a trip management into one common access point.

Project Motivation:

- Travelers currently use multiple applications to find itinerary resources, hotel reservations, and local tours and attractions. A unified service will help reduce the confusion and encourage environmentally-friendly tourism.

Project Approach:

- The Smart Tourism Platform will utilize an AI-based recommendation engine and geolocation data to create unique travel plans. The Platform will include global weather forecasts, information about upcoming events, ratings of sustainable tourism options, and listings of tours provided by local communities.

Applications of the Project:

The project will have multiple applications including,

- **Smart itinerary generation**--the system will be capable of generating an optimal travel plan based on user preferences;

- **Usage and promotion of local heritage sites**--the application allows users to explore local cultural heritage sites, and utilize the local tour operator to provide tours;
- **Creating awareness and education about sustainable tourism practices**--the project will promote sustainable travel through education, and

- **Provide a digital guide for rural tourism--**the solution incorporates a database of rural tourism information that can be accessed using mobile devices.

Limitations of the Proposed Approach:

- Requires consistent internet connectivity
- Data accuracy depends on real-time API feeds
- Limited adoption in areas with low digital literacy

Besides these limitations, the One Stop Solution for Tourism will be an effective solution for both business and leisure tourists.

1.5 Objectives

1. **Behavior:**
   Enable tourists to explore personalized, sustainable destinations based on preferences and past travel behavior.
2. **Analysis:**
   Collect and analyze travel data (visits, reviews, seasonal trends) to improve future recommendations.
3. **System Management:**
   Develop a centralized platform integrating booking, location, and sustainability modules.
4. **Security:**
   Ensure user data privacy through secure authentication and encrypted transactions.

5. **Deployment:**

   Deploy the solution as a **mobile + web app** with multilingual support for regional accessibility.

1.6 SDGs

The proposed system aligns with the following UN SDGs:

- **Goal 8:** Decent Work and Economic Growth – supports local tourism businesses.
- **Goal 9:** Industry, Innovation and Infrastructure – promotes digital tourism infrastructure.
- **Goal 11:** Sustainable Cities and Communities – encourages eco-friendly travel options.
- **Goal 12:** Responsible Consumption and Production – provides sustainable travel insights.
- **Goal 13:** Climate Action – promotes carbon-conscious travel planning.

1 NO POVERTY
2 ZERO HUNGER
3 GOOD HEALTH AND WELL-BEING
4 QUALITY EDUCATION
5 GENDER EQUALITY
6 CLEAN WATER AND SANITATION
7 AFFORDABLE AND CLEAN ENERGY
8 DECENT WORK AND ECONOMIC GROWTH
9 INDUSTRY, INNOVATION AND INFRASTRUCTURE
10 REDUCED INEQUALITIES
11 SUSTAINABLE CITIES AND COMMUNITIES
12 RESPONSIBLE CONSUMPTION AND PRODUCTION
13 CLIMATE ACTION
14 LIFE BELOW WATER
15 LIFE ON LAND
16 PEACE, JUSTICE AND STRONG INSTITUTIONS
17 PARTNERSHIPS FOR THE GOALS
SUSTAINABLE DEVELOPMENT GOALS

Fig 1.1 Sustainable development goals [1]

1.7 Overview of Project Report

The report presents the design and development of a digital platform referred to as the One Stop Solution for Tourism. This is intended to provide an integrated system for trip planning, booking and managing sustainable tourism activities, thereby improving the experience of traveling. The report introduced the background, motivation and objectives, and how this project aligns with the Sustainable Development Goals (SDGs) of the United Nations.

The report also provides a comprehensive and detailed literature review of existing tourism management systems including additional information and technologies used in the field of smart tourism, including a review of the inconsistencies in the current solutions available today. Finally, the report describes the proposed solution methodology, describes the system architecture and describes how the solution works by describing how the solution incorporates multiple modular components, including: (a) a user interface, (b) an itinerary management system, and (c) an application to allow for community-based traveller interaction. Discusses the software and hardware requirements, implementation details, and database design used to develop the system.

This concludes the report by summarizing the achievements of the project and emphasizing its contribution toward promoting sustainable and accessible tourism through digital innovation.

CHAPTER 2
LITERATURE REVIEW

The tourism sector is one of the most evolving and fast-growing industries today globally. Digital technologies have led to increased demand for solutions that are integrated to satisfy the varying needs of tourists. With that overview in mind, the goal of this literature review is to highlight the various technologies and methodologies that can be leveraged to provide a complete "one-stop" experience for tourism. One of the main objectives is to provide a more easily navigable, accessible, personalized and integrated service for tourists.

Borras et al.'s study is an analysis of how recommender systems (RS) for Tourism have been adapted and applied. They define four different approaches to constructing RS in Tourism: Rule-based; Collaborative, Content-based, and Hybrid. The article demonstrates that in order to create effective Itineraries for tourists, the context surrounding a tourist's location (time of year, weather, group) should be accounted for, as well as their experience (ratings, reviews, POI attributes).

Most evaluations of RS in Tourism are based on offline metrics (i.e., RMSE, Precision/Recall) but Borras et al. identify that there is a significant gap in measuring real-time Tourist satisfaction, Adoption of RS in Tourism and the Sustainability impact of RS on tourists. They recommend capturing richer context for tourists using mobile sensors and other social signals, and real-world evaluation of RS to address this research-practice gap. LIMITATIONS: The majority of the relevant research was published in 2014, so relatively few deep learning or large-scale mobile data studies are included in their review. Future Directions: Future research should re-evaluate the evaluation methods using the most recent literature relating to Deep models and Deployment of RS in Tourism.

The Literature Review of Literature 2024-2025 identifies the dominant categories for the most recent RS and the current focus on deep learning, contextually-aware systems, and cold start solutions for new destinations. Additionally, the Literature Review uses PRISMA-style selection criteria, demonstrating that hybrid approaches and knowledge graphs are better used for organizing the heterogeneous nature of tourism content (e.g., reviews, POIs, event feeds). The results of this study indicate that with current implementation of RS, data sparsity, privacy, and explainability issues remain; therefore, additional studies should be conducted for standardized benchmarking and field trials to provide empirical information on user experiences. Specifically, the limitation of many of the studies is that the vast majority of them

still depend on offline datasets; therefore, to improve future studies, researchers should work to create publicly available real-time datasets and standardized user-centric metrics. Surveys of IoT have shown that sensors, beacons, and connected devices will support the implementation of smart-city tourism services by providing developers with the tools necessary for real-time crowd monitoring, personalized, location-based content delivery, and smart infrastructure management. Several case studies highlight benefits from using IoT for tourism services, such as the ability to monitor the capacity of crowded locations in real time and provide visitors guidance based on contextual information, both of which reduce the risk of injury while improving the flow of visitors in a tourist area. Challenges that were associated with this was using IoT for tourism services include issues related to interoperability, costs associated with maintaining batteries, and the need for privacy when keeping track of a visitor's location. The paper proposes that implementing a federated architecture for all IoT geospatial data and implementing edge processing will help reduce latency issues and privacy-related risks, and that organizations should work with local partners to conduct pilot projects to demonstrate the potential value of these systems.

According to studies of augmented reality (AR) tour guides, using AR markers, navigation and other historical information on smartphone apps can lead to greater levels of engagement and learning when using an AR tour guide. Most studies on AR guide development create mobile AR applications and then test their effectiveness using actual users. Past studies have suggested that people use the AR guides more frequently, recall more things from their excursions to the area and enjoy using them. However, many issues must still be solved before mobile AR guides can become viable options for tourists. One significant challenge is the difficulties associated with creating content; for now, the majority of the effort put into AR content development is located mainly in the high costs associated with creating that content. In addition to content creation, there are also two significant types of localization challenges (GPS accuracy for outdoor areas, indoor location accuracy) and more significantly, the potential for users to become distracted when using an AR guide. The use of multiple resources is one way to improve mobile AR guides. Researchers have suggested using a crowdsourcing method to curate, or compile, AR tour guide data; use of both Bluetooth and geographic marker locations would enable users to create hybrid models of where they are; and careful UX design that recognizes the importance of balancing the density of information with the user’s safety. In addition to mobile AR guide development efforts, there has also been some research conducted using sentiment analysis and topic modelling. These efforts are directed at identifying patterns

of social media activity related to and impacting the perception of a tourist destination. Research to date indicates that data from social media may provide an early warning system for emerging issues (ex. customer service complaints; safety incidents); and additionally offer insights into the ongoing changes that occur with regard to the geographical location of a destination over time. To conduct analyses, researchers have employed techniques such as lexicon-based and transformer-based classifiers for content categorization using Twitter, Instagram and TripAdvisor content. By combining this data with geotagged posts and statistical time series data researchers can examine seasonal variances and events that impact perceptions of destinations. Limitations associated with these analyses are the existence of biased samples (only a small portion of all tourists will post something) and the nature of social media itself (noisy language) as well as non-representative geospatial coverage. Improvements in future analyses may include fusing multiple platforms (Twitter, Instagram, TripAdvisor) to create a bigger sample, developing multilingual classifiers and performing ground truth verifications.

Recent literature reviews have shown how digital technologies (big data, IoT, AI, mobile technologies) are used in the development of smart tourism through the way they are structured and managed. The literature shows a cluster of themes related to the development of smart tourism: personalization of the visitor experience; development of destination management platforms; integration of mobility and transportation; and development of sustainability monitoring systems. The most consistent gap noted in studies of smart tourism is the lack of integration across the three disciplines of technology, policy, and community. The authors suggest that in order to support equity in the distribution of benefits from smart tourism, there should be co-creation with local communities; inclusive design of smart tourism systems, and the establishment of evaluation frameworks to assess social and economic impacts on local communities.

New research suggests the use of context-aware recommender systems to optimize for sustainable tourism objectives (low-carbon route recommendations, community friendly experiences). The recommender systems are implemented using deep neural networks and context-based encoders. The trade-offs between maximizing user satisfaction versus minimizing environmental impacts are evaluated using multiple-objective optimization methods. The results demonstrated that it is possible to induce tourists to take greener options with minimal sacrifice in utility if they are provided with sufficient explanations and separation incentives (discounts or badges). Limitations in the available literature include difficulties in

obtaining sustainable tourism scores, reliable environmental footprints for points of interest (POIs), and ways to operationalize carbon output calculations. The authors suggest that improvements can be made by integrating with existing carbon calculators and local carrying capacity sources.

By using TAM (Technology Acceptance Model) as the basis for empirical research, researchers examined the effect of perceived ease of use, perceived benefit, and perceived trust in regard to the adoption of travel applications, along with social influence and risk perceptions. Structured surveys using field research and structural equation modelling revealed perceived trust and perceived data security strongly correlated with the adoption of travel applications, particularly in the case when payment and ID input were involved. Some limitations of the current research are the potential for cultural biases, the sample population predominantly being students, and the limited follow-up after the point of initial longitudinal support. Improvements to the research include longitudinal studies on the continued use of Travel Applications and testing of alternative layouts for their User Interfaces through A/B testing in order to measure retention and conversion rates.

There have been several systematic and comparative-case studies conducted which demonstrate how these Digital Platforms have the potential to facilitate Sustainable Tourism by facilitating distribution of visitor flows, encouraging travel to less visited periods, and also enabling local Micro-entrepreneurs to sell goods and services. The empirical evidence has demonstrated that Digital Nudges and Curated Itineraries can diminish the effects of overtourism on specific Destinations and redistribute the Collection of Revenue into Periphery areas. However, these outcomes depend heavily on Stakeholder Collaboration, the availability of accurate Local Data, and incentives for Local Businesses to participate in Digital Platforms. The continuation of these models requires the creation of co-managed Destination Platforms, Revenue Sharing Models for Local Hosts, and the establishment of Success Metrics for both Economic and Environmental Impacts.

Summary of Literatures reviewed

Table 2.1 Summary of Literature reviews

Paper / Topic (short)	**Year**	**Method**	**Main finding**	**Noted limitation**
Tourism RS foundations (Borras)	2014	Survey of RS types	Context + hybrid RS needed for usable itineraries.	Few field evaluations.
RS systematic review	2024	PRISMA SLR	Deep/context-aware RS emerging; hybrid, KG methods.	Data sparsity; lack of benchmarks.
IoT in tourism	2019–2022	Literature + case studies	Real-time monitoring & personalized services possible.	Privacy, cost, interoperability.
AR tour guides (prototype)	2018–2024	App dev + field eval	AR improves engagement and learning.	Content creation cost; localization accuracy.
Blockchain frameworks	2020	Framework + interviews	Can increase transparency/payments efficiency.	Scalability, regulation, energy.
Social media sentiment	2021–2024	Sentiment/topic analysis	Useful for real-time destination image monitoring.	Sampling bias, noisy text.
Smart tourism review	2024–2025	SLR/state-of-art	Digitalization underpins smart tourism; sustainability link.	Few integrated socio-technical deployments.
Context-aware sustainable RS	2025	Deep models + multi-objective	Can nudge toward low-impact options with minor UX cost.	Hard to compute accurate sustainability footprints.
Travel app adoption	2022–2025	Surveys, SEM	Trust and perceived security = strong adoption drivers.	Sample bias; limited longitudinal data.
Digital platforms → sustainability	2024–2025	Case studies + SLR	Platforms can redistribute demand and support local economies.	Stakeholder coordination required; data

CHAPTER 3
METHODOLOGY

Tourism platforms need rapid user-facing features (recommendations, itineraries, bookings) plus reliable backend services (maps, payments, sustainability metrics). Agile Scrum gives iterative delivery and stakeholder feedback. DevOps adds CI/CD, automated testing, scalable deploys and monitoring — essential for a production tourism platform. The hybrid keeps development flexible while ensuring operational reliability.

High-level lifecycle mapped to standard SDLC stages

1. Requirement Gathering & Analysis (Specification, literature)

- Activities: stakeholder interviews (tourists, guides, local vendors, tourism board), competitive analysis, literature review (recommender systems, IoT for crowd sensing, AR guides), data sources inventory (POI datasets, weather APIs, event feeds).
- Deliverables: Product Vision, Requirement Backlog (user stories), Non-functional requirements (latency, uptime, security, GDPR), Data & API specification.
- Acceptance: Backlog prioritized with business value & risk; data sources listed and sample agreements verified.

2. System and Functional Architecture Design (Architecture, Modules)

- Activities to perform to define System Architecture (mobile/web front end, backend services, recommendation engine, geolocation/map type modules, payment modules, admin panel) as well as choose Data Models to use; API definitions/security/privacies including (authentication and encryption).
- Deliverables produced as a result of architectural creation includes Developing High-Level Architecture Diagrams; Component Diagrams, API Contracts, Data Schema and Sequence Diagrams for each business' core flow (from booking to itinerary production).
- Acceptance was obtained through methods including Architectures Reviews with Stakeholders; Scalability and Failover Strategies that had been approved.

3. Unit/Component Design of Modules (Hardware and Software):

- Activities: to create detailed specifications of each module would include features of Recommendation (i.e., feature design) as well as if the modules include IoT

Ingestion (i.e., if used), Provide Asset Pipelines for surfaced augmented realities as well as to have a Database Schema with Microservice Contracts between Hardware and Software types and specify the Firmware and Interface Types involved by all Kiosk or Beacon Devices fulfilled by end user devices.

- Deliverables: produced Affirmed that Module Specifications would be required, Pseudo Code/ Flowcharts will be produced, and Hardware Interface Specification will be documented.
- Acceptance: will occur through reviews of each Module Specification by Peer Network members, via approvals at Point of Design Signatures.

4. Implementation (Code and Unit Testing):

- Activities will occur based on a Sprint Capacity for Implementation of User Stories where Developers must create Unit Tests for the code they develop; Create Inline Comments for their Code as well as Code Review by others.
- Deliverables Affirmed as Completed include Working Branches with features under development; and Unit Test Suites created(e.g., 70%-80% Test Coverage).
- Acceptances will occur when Unit at all Test occurs where no failure, and PR reviewed as Approved.

5. Integration and System Testing:

- Activities: Services are integrated within the test environment and integration tests (API contract test, end-to-end flows) are executed. Data pipeline checks performed, security scanning completed.
- Deliverables: Test Reports (Functional and Integration) with defects and fixes listed.
- Acceptance Criteria: No Critical defects present and passed Regression Testing.

6. Verification and Validation (V&V).

- Verification verifies that the system is built per the guidelines received (design review, code audit, automated CI checking).
- Validation validates that the system fulfills the requirements of the stakeholders (user acceptance testing, pilot testing with small groups of tourists, validating the system's functionality and capabilities against real-world situations).
- Outputs: User Acceptance Test (UAT) approval, utilization evaluations, performance benchmarks.

- Requirements for Acceptance: UAT approval, followed by a decision to go/no-go for Production Release.

7. Deployment and Release (DevOps).

- Tasks: Implement automation of deployments via a Continuous Integration/Continuous Deployment (CI/CD) pipeline; Containerization of services via Docker, orchestration of Containers (Kubernetes or managed services); creation of monitoring and alerting systems; and development of Backups and Rollback Plans.
- Outputs: Deployment Scripts; Run Books; Monitoring Dashboards; Service Level Agreement (SLA) Documentation.
- Requirements for Acceptance: Successful execution of Smoke Tests in Production; Clearly defined Monitoring Baselines.

8. Maintenance and Continuous Improvement.

- Tasks: Monitor live data; Utilize A/B testing experiments for Recommendations; Collect feedback from Community members; Manage Backlogs for Iterative Development.
- Outputs: Release Notes; Incident Reports; Roadmap Updates.
- Requirements for Acceptance: Meeting SLA targets; Consistent Martin and affiliates Backlog Grooming.

Agile Scrum specifics (roles & artifacts)

- **Roles**
  - Product Owner (tourism stakeholder or project lead) — defines priorities, acceptance criteria.
  - Scrum Master — removes impediments, enforces sprint cadence.
  - Dev Team — frontend, backend, ML engineer, QA, DevOps engineer.
  - UX/Content Manager — handles AR content and local vendor onboarding.
  - Stakeholder Panel — local tourism board, sample tourist users, vendor reps.
- **Artifacts**
  - Product Backlog (user stories prioritized by value/risk)
  - Sprint Backlog (2–3-week sprints recommended)
  - Definition of Done (DoD: code + unit tests + doc + security scan)

- Increment (potentially shippable feature)

- **Ceremonies**

  - Sprint Planning, Daily Stand-up, Sprint Review (demo to stakeholders), Sprint Retrospective.

DevOps & CI/CD pipeline (core activities)

- **Source control**: Git (main, develop, feature branches).
- **Build & Test**: CI server (GitHub Actions / GitLab CI / Jenkins) runs unit tests, static analysis, security scanners.
- **Containerization**: Docker images for services.
- **Deployment**: Kubernetes / cloud managed services (EKS/GKE/Azure AKS) or serverless where suitable.
- **Monitoring**: Prometheus + Grafana, Sentry for errors, ELK stack for logs.
- **Autoscaling & resilience**: use health checks, horizontal pod autoscaling, circuit breakers.
- **Secrets**: vault (Hash Corp Vault or cloud KMS).
- **Backup & rollback**: automated DB backups, blue/green or canary deploys.

Testing & V&V plan (mapped to SDLC)

- **Unit testing** — developers (each module), target ≥70% coverage.
- **Integration testing** — service contracts, API correctness.
- **System testing** — end-to-end flows (search → itinerary → booking).
- **Performance testing** — load tests for peak tourist events (use JMeter / k6).
- **Security testing** — static code analysis, dependency scans, OWASP ZAP for web testing, penetration testing prior to launch.
- **Usability / UAT** — usability testing sessions with small tourist groups, heuristic evaluation.
- **Acceptance testing** — stakeholders sign off on acceptance criteria.
- **Regression testing** — automated test suite on each CI build.

Data, Privacy & Security (important for tourism apps)

- Use **OAuth2** or OpenID Connect for authentication.
- Encrypt PII at rest and in transit (TLS + DB encryption).
- Offer privacy settings and explainable data usage for users (consent flows).
- Anonymize analytics and follow local regulations (e.g., GDPR if EU users).
- Maintain PCI compliance for payments or integrate PCI-certified gateways (Stripe/PayPal).

Tools & Technologies (recommended stack)

- **Frontend**: React Native (mobile) or React (web).
- **Backend**: Node.js (Express/Factify) or Python (FastAPI).
- **Database**: PostgreSQL (primary), Redis (cache), Elasticsearch (search).
- **Maps & Geo**: OpenStreetMap + Leaflet or Map box for advanced styling.
- **Recommender/ML**: Python (scikit-learn, PyTorch/TensorFlow), or integrate lightweight microservice for recommendations.
- **CI/CD**: GitHub Actions.

Components to include

1. User devices (Mobile app, Web app, AR headset)
2. API Gateway / Load Balancer
3. Auth Service (OAuth2)
4. Recommender Service (ML model + features store)
5. POI & Metadata DB (Postgres) + Search (Elasticsearch)
6. Booking & Payment Service (integrated gateway)
7. IoT Ingestion (if used) → Stream Processor (Kafka) → Analytics DB
8. Admin Panel / Content Management
9. Monitoring & Logging (Prometheus/Grafana / ELK)
10. Third-party APIs (Weather, Maps, Event Feeds, Social Data)
11. CDN & Static Assets (images, AR content)

Draw.io steps

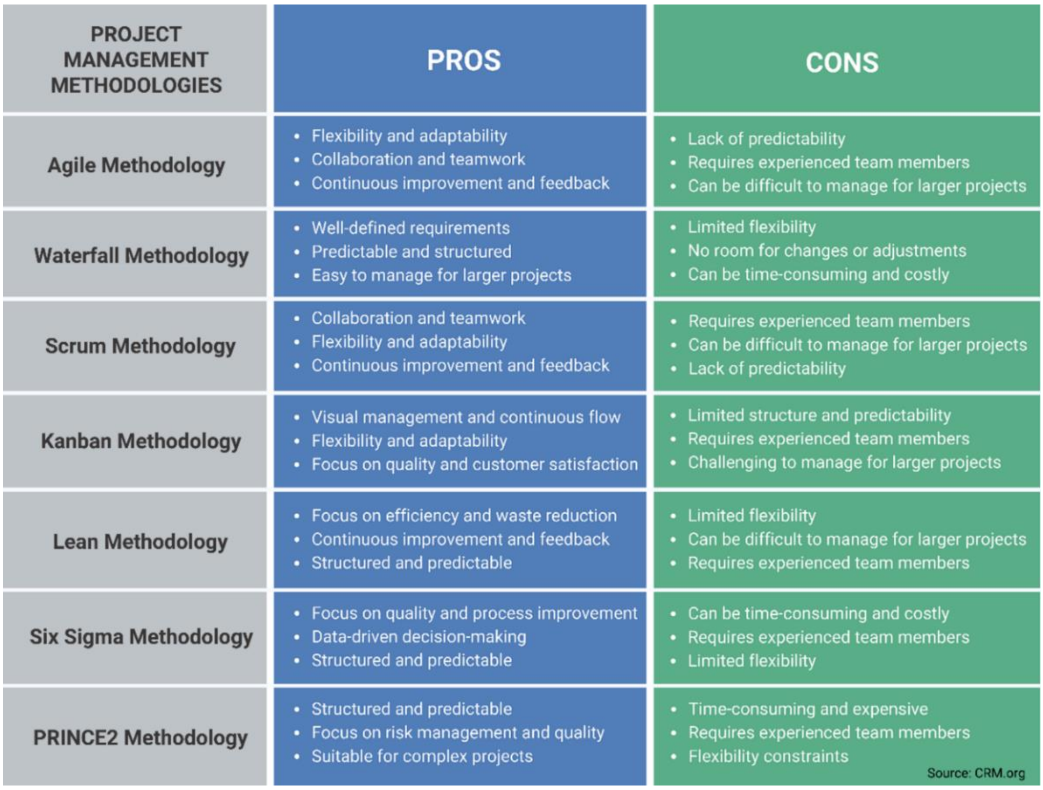
- Open Draw.io → New diagram → choose “Blank”.
- Use “Rectangle” shapes for services, “Cylinder” for databases, “Cloud” for external APIs.
- Place “User devices” at left, “Third-party APIs” at right. Connect arrows to API Gateway → microservices → DBs.
- Label arrows with protocols (HTTPS, MQTT, WebSocket).
- Export as PNG/SVG for embedding in the report.

Deliverables (for your report)

- Product Vision & Requirement Backlog.
- Architecture diagram (Draw.io output).
- Module design docs.
- CI/CD pipeline definition and runbook.
- Test plan & test reports.
- Source code repo and deployment manifests.
- Final report with sustainability alignment (SDGs) and impact metrics.

Risk management & mitigation

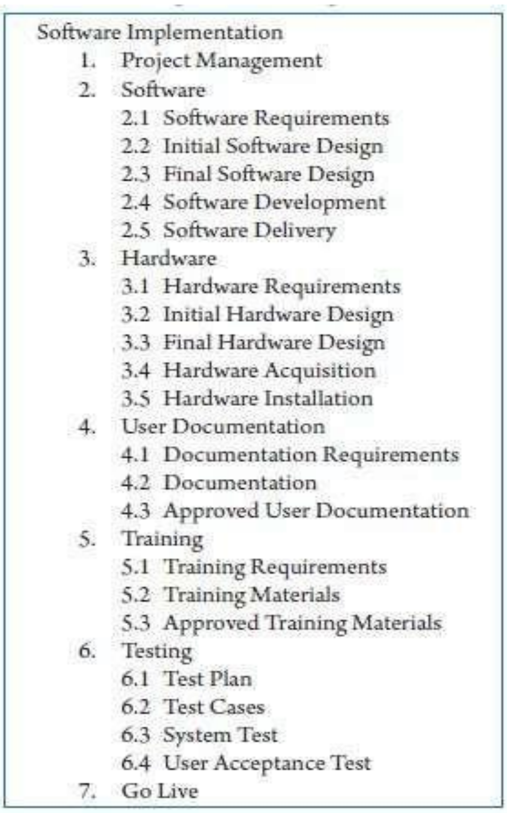
- **Data privacy risk** → mitigation: privacy-by-design, consent, encryption.
- **Third-party API downtime** → mitigation: caching, graceful degradation, fallback data.
- **Scalability under events** → mitigation: autoscaling, load testing scenarios.
- **Local adoption** → mitigation: co-creation with local stakeholders, multilingual UX, low-bandwidth mode.
- **Content accuracy** → mitigation: vendor verification flow and crowdsourced moderation.



PROJECT MANAGEMENT METHODOLOGIES	PROS	CONS
Agile Methodology	• Flexibility and adaptability • Collaboration and teamwork • Continuous improvement and feedback	• Lack of predictability • Requires experienced team members • Can be difficult to manage for larger projects
Waterfall Methodology	• Well-defined requirements • Predictable and structured • Easy to manage for larger projects	• Limited flexibility • No room for changes or adjustments • Can be time-consuming and costly
Scrum Methodology	• Collaboration and teamwork • Flexibility and adaptability • Continuous improvement and feedback	• Requires experienced team members • Can be difficult to manage for larger projects • Lack of predictability
Kanban Methodology	• Visual management and continuous flow • Flexibility and adaptability • Focus on quality and customer satisfaction	• Limited structure and predictability • Requires experienced team members • Challenging to manage for larger projects
Lean Methodology	• Focus on efficiency and waste reduction • Continuous improvement and feedback • Structured and predictable	• Limited flexibility • Can be difficult to manage for larger projects • Requires experienced team members
Six Sigma Methodology	• Focus on quality and process improvement • Data-driven decision-making • Structured and predictable	• Can be time-consuming and costly • Requires experienced team members • Limited flexibility
PRINCE2 Methodology	• Structured and predictable • Focus on risk management and quality • Suitable for complex projects	• Time-consuming and expensive • Requires experienced team members • Flexibility constraints

Source: CRM.org

Fig 3.7 Summary of various methodology



Software Implementation

1. Project Management
2. Software
   - 2.1 Software Requirements
   - 2.2 Initial Software Design
   - 2.3 Final Software Design
   - 2.4 Software Development
   - 2.5 Software Delivery
3. Hardware
   - 3.1 Hardware Requirements
   - 3.2 Initial Hardware Design
   - 3.3 Final Hardware Design
   - 3.4 Hardware Acquisition
   - 3.5 Hardware Installation
4. User Documentation
   - 4.1 Documentation Requirements
   - 4.2 Documentation
   - 4.3 Approved User Documentation
5. Training
   - 5.1 Training Requirements
   - 5.2 Training Materials
   - 5.3 Approved Training Materials
6. Testing
   - 6.1 Test Plan
   - 6.2 Test Cases
   - 6.3 System Test
   - 6.4 User Acceptance Test
7. Go Live

Fig 3.9 Summary of project breakdown to task

CHAPTER 4
PROJECT MANAGEMENT

4.1 Project timeline

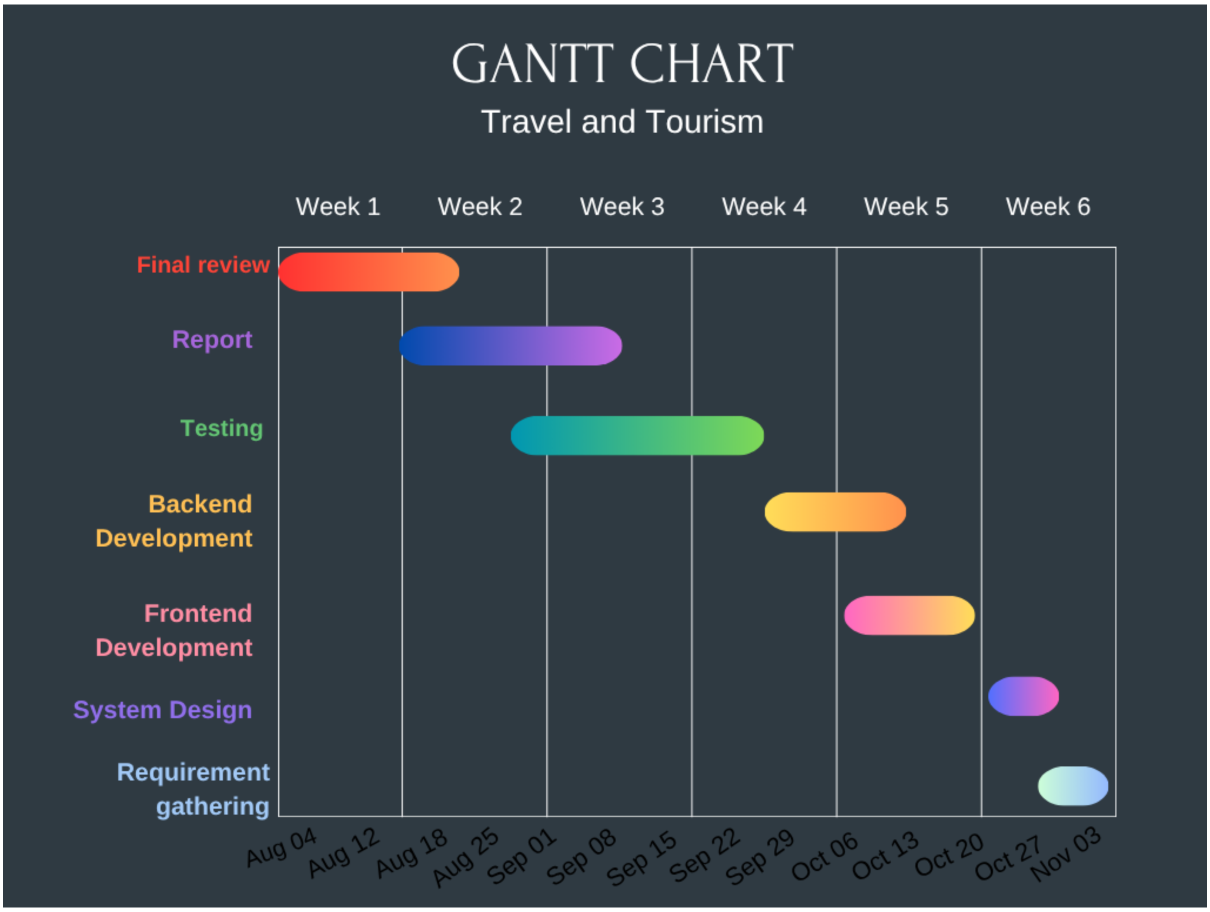


Fig 4.1 Project implementation timeline

4.1.1. Requirement Gathering & Feasibility Study

- Establish the goals of the tourism platform as well as its intended users
- Use surveys, interviews, existing travel websites analysis as methods used to gather user requirements
- Assess whether or not there is a technical possibility to integrate multiple tourism service types; include a review of current technologies available
- Determine potential hardware, software, and technical limitations

4.1.2. Create the System Design: UI, Database, & Architecture

- Utilize an HTML wireframe to illustrate how the tourism web portal is organized (default layout) and how navigation flows through the pages
- Create a database schema for storing user profiles, user accounts, bookings, reviews, and all other user-generated content
- Identify a modular architecture approach (for example: Service Provider modules, Admin Modules, End User Modules)
- Create ER diagrams and detailed logical design documentation for future development

4.1.3. Create the Frontend Using HTML, CSS and JavaScript

- Create the front end of the site using HTML and CSS to generate and publish responsive design webpages
- Use JavaScript to provide interactivity with webpages (i.e., dynamic forms, filters for searches) creating a more engaging experience for users
- Ensure that the site is compatible with all major web browsers and that users are able to easily navigate through all site content and functions
- Validate input by users; provide users with error messages when inappropriate or invalid data is entered

4.1.4 Create the Backend & Integration of APIs and Payments

- Create backend functionality to manage and maintain all user logins, registrations, and bookings
- Integrate APIs for Hotel Booking, Transportation Scheduling, or any other type of Tourism Information where applicable
- Configure the payment gateway and procedures for securely processing and maintaining sensitive data
- Ensure that all API calls function as intended and that proper communication happens between the front and back end

4.1.5. Testing and bug fixing

- Unit testing for individual module and integration testing for the combination of the modules
- Recognize and rectify UI errors, logical mistakes, and validation problems.
- Verify that functionality works as per the requirements.
- Make sure that your application has good error handling and is durable across various scenarios and user interactions.

4.1.6 Documentation & Report

- Create a technical overview of how your application works, the code structure and design of your application.
- Add screenshots, user manuals, and a diagram showing the database structure to the overview.
- Combine the results from the tests and the performance metrics into one document. This document is intended for internal use.

4.1.7 Final Review - Will Include - Deployment - Presentation

- Host the production instance on local or remote server for demonstration purposes
- Go through a complete review of your application with your mentor as a finished product.
- Prepare presentation slides and video demonstration.
- Submit final report and make final capstone presentation.

4.2 Risk analysis

4.2.1 Technical Risk

- Description: Due to integration of Multiple Tourism Service (Hotel Booking + Destination Search + Payment System), The risk of compatibility issues between modules or APIs are high as they would be dependent on each other.

- Impact: Due to potentially mal function of a key tourist feature (Saving of Searches, completed booking, etc...) could result in lost sales.
- Mitigation: Regularly conduct integration tests, maintain highly modular code and use only proven stable APIs & Frameworks contribute to reducing technical risk.

4.2.2 Security Risk

- Description: The Platform stores and manages sensitive customer information (Name, Email ID, Route Information, etc...) making it a target for or vulnerable to being hacked or had unauthorized access to.
- Impact: Could cause Data Loss, Server Downtime, and/or ultimately result in User Loss of Trust.
- Mitigation: Implement HTTPS Encryption for transfer of information, Input Validation for Data Integrity protection, secure Authentication for accessing the system and manage session properly.

4.2.3 Data Risk

- Description: Data Loss may occur due to Software Error, Hardware Error or by improper handling by Developer during application update process.
- Impact: Unpredictable Booking Missed Commencement Dates could disrupt user bookings, Searching for Hotels within the booking population, or having no records available for searching or booking of hotels.
- Mitigation: Implement strict schedule for conducting Regular Database Backups and utilize Cloud Based Storage with proper Version Control.

4.2.4 Time Management Risk

- Description: Project has multiple stages of Development; that is Front End Development, Back End Development, Testing, Deployment.
- As each stage of Development is dependent on the other, Delay of one will effect all stages and therefore, the timeline.
- Impact: Without strict adherence to the Gant Chart, Review of Timelines weekly with Team Members, assign specific responsibilities to specific Team Members to ensure Timely completion of all Development Stages.

4.2.5. User Interface (UI/UX) Risk

- This outlines potential risk due to a poorly designed user interface and slow response times for visitors on the website.

- As the website users are not engaged with the website, this could lead to a reduced number of users and/or less satisfaction with their experiences.
- Mitigating this risk requires performing usability tests, optimizing the website code for performance, and following the principles related to responsive design.

4.2.6 Maintenance and Scalability Risk

- This section describes the potential risk posed by the growth of the user base on the website.
- Challenges with scaling and supporting a higher volume of visitors may arise from the growth of users in the future, creating an increased level of performance and availability risks for the website.
- Ways to mitigate performance and availability specific risks are to optimize the website's code, maintain a modular site architecture (e.g., breaking the website into separate modules of functionality) and plan for adjustments in the future.

4.2.7 Ethical and Legal Risk

- This section outlines the potential ethical and legal risks associated with using copyrighted images, information, and inaccurate data regarding travel.
- Should legal action be taken against the website for copyright infringement, this would lead to the possibility of legal disputes and penalties.
- Ways to mitigate these potential legal issues are to use royalty-free assets, provide specific citations for the images and information used, and ensure that all displayed information is transparent.

4.3 Project budget

1. The first part of our plan was to break up the project into five separate smaller tasks, as determined by our project objectives. The creation of each task allowed us to develop a Gantt chart showing everything from the creation of our website through to its completion. Out of this initial breakdown of the overall project, we identified a total of 5 primary tasks:

    1. Requirement Gathering & Feasibility Study.
    2. System Design (User Interface, Design, Database, and System Design).
    3. Frontend Development (HTML/CSS/JavaScript).

4. Backend Integration (APIs, Payment Integration)

. 5. Testing and Bug Fixing.

6. Writing the Required Documentation and Report.

7. Final review, Deployment, and Presentation of the project.

2. From there, we looked at the availability of our team and compared this with our Project Timeline, ensuring that it was realistic to carry out the project in the time frame we were working with. We compared the planned project events against the calendar of our free time and any other academic commitments, including our studies and work placements. The individual tasks assigned to each member of the team were also allocated based on familiarity, and our previous experience, to enable us to balance the workload efficiently and to adhere to realistic deadlines.

3. In order to calculate the time required for each of the 5 tasks, we will first break it down into smaller, manageable tasks; for example, Task 1 will now consist of smaller sub-tasks, such as 'File Permission Check'. Once each sub-task was identified, we will then proceed to estimate the time required for each sub-task and then report back the total estimated time for all 7 tasks on the Gantt Chart.

4.Our estimates in time were determined based on the experience of everyone on the team in regards to their classes discussing JavaScript, operating systems, and principles in software development. In addition, due to the direct involvement of some members of the team in certain areas of the project, we had a clearer idea of the amounts of time that would be required. Because of the research and the learning curve associated with some areas of the project, we built in a buffer on the estimated times for each task. This enabled us to develop an aggressive but realistic timeline for the project

5. Directly, the financial aspect (budget) for this project is zero Rupees (₹0.00). This was accomplished through the thoughtful use of free resources, as well as the members of the project team's current skills. In lieu of any monetary expense, a thorough resource plan was still important to have. The primary "cost" incurred was the time invested by the project team, which could be measured in man-hours.

6. Since the financial budget was zero, we concentrated on our greatest resource, time, for tracking purposes. We utilized the Gantt chart and the project plan to compare the actual progress vs. the planned deadlines. The primary method of "budget" tracking was to ensure that the actual number of man-hours used by the project team were not significantly more than

our original estimates for the task, thereby allowing us to stay on schedule for the project.

Table 4.2 Example of project budget

Resource Category	Item	Estimated Cost (INR)	Justification / Notes
1. Personnel (Labor)	Project Team Labor	₹0.00	The project was developed by the team as part of our academic curriculum. All labor was contributed voluntarily.
2. Software	Python, VS Code IDE, Git	₹0.00	All development software used is opensource and available free of charge.
	Windows 11	₹0.00	Operating systems were used under existing academic or personal licenses.
3. Hardware	Development Laptops	₹0.00	We utilized the project team's existing personal computers for all development and testing.
4. Hosting & Collaboration	GitHub Repository	₹0.00	The project code and documentation are hosted on GitHub's free tier for opensource projects.
Total Financial Cost		₹0.00	

If this study has been undertaken commercially, the increased commercial development costs associated with that are as follows:

- Premium API Subscription: Higher volumes of traffic.
- High availability, performance, and scalability through a dedicated server infrastructure.
- High-quality software licenses for testing tools, project management tools, etc.
- Marketing and sales activities for product promotion.
- Salary expenses associated with a dedicated development/support team.

Overall, the total amount spent on increased costs as a result of commercial development was within the scope of managing this academic project. The most significant contribution to this study was the amount of time and intellectual resources provided/expended by the project team.

CHAPTER 5
ANALYSIS AND DESIGN

The " One Stop Solution for Tourism " will be a comprehensive resource that combines numerous services related to tourism into one complete trip. The design of the system includes two entrances into the program — a web-based entrance and a mobile entrance, which will allow users to access the programmer in a manner most convenient to them (whether it be desk or mobile).

Additionally, as part of the development process of the application, there will be three different 'modules' for the application: User Auth Module, Database Module and Service Integration Module. By developing each of these modules separately, the application will become easier to scale as well as to maintain.

As an example, the application can take advantage of Cloud Computing services to store and process data in a safe manner using APIs (Application Programming Interfaces) to connect with the various service providers (e.g. airlines, hotels, attractions) and provide a mechanism for checking their availability and prices in real time. Finally, the creation of a community of travelers who contribute content and reviews about their experiences will create a more engaging and beneficial format for decision-making for future travelers.

The "One Stop Solution for Tourism" will present a holistic platform where diverse tourism services are integrated into an all-i5nclusive travel experience for users. System Design A user-friendly web and mobile interface will be available in user access, including itinerary planning, making a reservation over accommodation, purchasing local attractions or guided tours, and transportation options.

The implementation should be modular. This will disintegrate main functionalities such as user authentication, database management, and service integration and make it easily scalable and manageable.

The core of the app can use cloud-based services to safely store and process its data, with APIs used in communication with services from airlines and hotels, hotels, and regional attractions for a real- time checking of availability and price. Furthermore, adding user-generated content and reviews will promote a dynamic community, where travelers can make the right choices.

This will allow for the ease of booking a single trip from multiple different sources while promoting and building brand loyalty through ease of access. By providing their customers with a recommendation engine that will enable them to see what their past behavior was and based on that, create an itinerary with tailored recommendations that will provide those customers with an enhanced tourism experience. This platform is designed with both personal privacy and data security as its main goal. We will implement high-quality security measures (e.g., encryption protocols), and be compliant with GDPR and other data protection laws. Therefore, consumers can build confidence and trust in our services because they know their information will be secure.

Additionally, we will allow our customers to benefit from the collaboration of local tourism boards with local tourism services to offer consumers travel packages and special offers as well as generating income and providing promotional opportunities to local businesses.

With this complete approach, we will create an unparalleled travel experience and support sustainable tourism by providing consumers with access to areas that are not typically explored. Essentially, this One Stop Solution for Tourism will simplify planning, promote cross-cultural exchange, and further local economies of destination.

5.1 Requirements

In this phase, our team captured the system's purpose, behavior, and detailed requirements.
The core objective is to create an automated tool to simplify and standardize security compliance checks.

System Hardware Requirements Phase

1. **Identify initial conditions**
   - Host on cloud provider (AWS/GCP/Azure) or hybrid cloud.
   - Optional edge devices: kiosks at tourist sites; BLE beacons for indoor localization.
2. **Determine input parameters**
   - Mobile/web requests, IoT beacon signals, third-party API data (maps, weather), vendor availability feeds.
3. **System outcome**

- Low-latency API responses; secure payment processing; accurate geo-services.

4. **Formulate relations**
   - User → Recommender → Itinerary → Booking → Payment → Ticket issuance.
5. **Identify constraints**
   - Bandwidth, device battery for beacon interactions, cost budget for edge hardware, physical installation permissions.
   - **Recommended HW**

- Cloud VMs / managed K8s cluster, CDN for static assets.
- Optional: Raspberry Pi kiosks, BLE beacons, edge processors for local analytics.

1. **Initial conditions**
   - Multi-tenant service, multi-region deployments for latency.
   - Support multiple languages and currencies.
2. **Input parameters**
   - JSON REST/gRPC calls, WebSocket for push, MQTT for IoT, payment gateway webhooks.

System outcomes Table 5.1 Summarizing requirements

Test level	**Who**	**Purpose**	**Acceptance criteria**
Unit tests	Devs	Validate single module logic	All unit tests green; coverage ≥70%
Integration tests	Devs/QA	API contract & external integration	No critical failures; webhooks & 3rd-party mocks pass
System tests	QA	End-to-end flows (search→book→pay)	End-to-end success for core flows
Performance tests	DevOps/QA	Load & stress under peak	p95 latency <200ms; sustains expected concurrency
Security tests	Security team	OWASP checks & pen test	No high-severity vulnerabilities

- Secure auth, reliable booking, consistent state across microservices.

3. **Formulate relations**
   - Microservices communicate via API gateway; message bus (Kafka) for async events.
4. **Constraints**
   - GDPR/region privacy laws, PCI compliance for payments, mobile offline capability.

5.1 Block Diagram

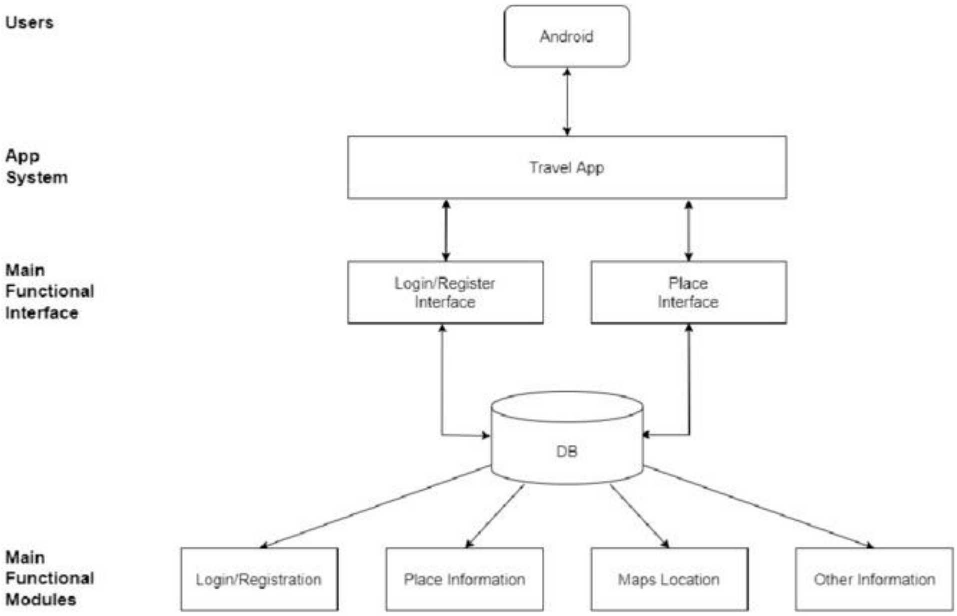


Figure 5.1 Block Diagram

1. Layer for Users

- The component of this layer is Android (End User Device)
- This layer is represented by the users of the system which can either be tourists, or travel enthusiasts, who use their Android device, such as a smartphone, to interact with the application.
- The end users of the Android Application (the Travel App) have the ability to:
  - ❖ Register/Login;

- Search for Tourist Attractions;
- Access maps and locate directions;
- and view all other areas of interest to the tourist, including Hotels, Vouchers to Attractions, Transportation, etc.

2. Layer for Application System

- The Travel App Application system is the core layer of the Application that acts on behalf of the User Interface and provides communication to the Back End or database system
- And all the other functions that this application performs are built as separate modules or interfaces;
- Therefore, the application itself is the Travel App and the user's first point of contact is either the Login/Register interface or the Place Interface, and the Travel App serves as the middleman between the user's request(s) and Database Operations.

3. Layer for Main User Operations Interface

The Main User Operations interface contains two main interfaces that are responsible for managing different types of User Operations.

(a) Login/Register interface — includes all user **Authentication and Registration** functions such as: Verifying the user's identity, storing the user's Login Credentials and Session Data and interacting with the database to retrieve or update the user's records.

(b) Place Interface — Manages user requests for travel destinations/tourist attractions. The Place Interface retrieves the information about Places from the Database, including the name, location, description, nearby facilities, images, etc., and it may also utilize the Map API to show the user where the place is as well as directions to the user.

4. Database DB

- The database serves as the main part of the app & backend architecture.

- It is responsible for storing and organizing the major data types that the app requires to function such as:

- ❖ User profiles and Login credential
- ❖ Information about tourist locations and place names
- ❖ The apps maps/routes/navigation information
- ❖ And all other associated types of data such as event listings, reviews, and vendor information.

- The database is the central point of reference for all app modules/interfaces. When an app module/interface needs to read from or write to the database, it performs these actions through a defined interface that connects to the database.

5. Main Functional Modules Layer

The main functional modules layer consists of the main modules that contain the application functionality. Each of the modules can communicate with the database to retrieve the data it requires or to store the data it is creating.

(a) Login/Registration Module:

- The login/registration module allows users to authenticate (login/logout) themselves, create an account, and recover their passwords.
- This module is responsible for maintaining the privacy of users; data and for protecting against unauthorized access.

(b) Place Information Module:

- The place information module contains details on all of the tourist places (attractions, days/times of operation, entry cost, etc.) that the app covers.
- This module could also include images, reviews/ratings, and local recommendations on the various tourist locations.

(c) Maps Location Module:

- The maps location module provides access to Google Maps (or alternatives) to show users the exact location of all tourist sites.

- In addition to providing users with a view of the map, this module will also provide users with navigation routes to get to (and from) a tourist attraction, as well as the estimated time for each leg of the route to/from the tourist attraction.
- The module will also allow users to find services (such as nearby hotels, places to eat, and ATMs) near their chosen tourist site.

(d) Other Information Module:

The other information module provides access to further local information, including:
• Upcoming cultural events.

• Weather information.

• Emergency contact numbers for hospitals/police/transportation services etc.

• Travel tips/sustainable tourism advice.

6. Data Flow Summary

The steps involved with executing requests through our Travel Application.

1 - The User launches the Application (Android).

2 - The Travel Application sends the request made (or taken) by the User to the proper interface: either the Login/Register or Place Interface.

3 - The selected interface queries Database (DB) to retrieve the information it needs to serve

the User.

4 - The Response is returned to the User Interface from the DB.

5 - The User understands or interprets their results through the Travel Application on their Android phone.

7. Objectives of the Architecture

•The purpose of showing how the structure of the Travel Application is designed to operate in a Modular and Data-Based fashion.

• Interfaces act as a conduit between User-Facing Components and Back-End Database.

• The objectives of the Travel Application Architecture are as follows.

o Provide clear separation of functionalities (Login, Places, Maps, Information) into 4 Modules.

o Scalable, providing for new modules to be added to continue enhancing the Travel Application.

o Secure, separating travel data from login data (so that even in the event the Travel Data is

compromised, the login remains protected).

o Efficient, as the Database is a Centralized Repository for all Travel Information.

5.2 System Flow Chart

Description of System Flow Chart: This flowchart depicts the overall flow of the Travel

Application's execution.

1. Traveler

- The Traveler is the primary end-user for the Travel Application.
- The Traveler will access the Travel Application via the Tourism website.
- After browsing Destinations, booking Accommodations and/or Travel Packages, Paying, and Providing Feedback after travelling, the Traveler has directly interacted with the Website using a user-friendly interface created in HTML/JavaScript.

2. Administration

- The Administrator is responsible for the management of the entire tourism platform and monitoring of service offerings to Users.
- Responsibilities of the Administrator include the ability to add or update Destination Information, verify Service Provider Accounts, manage User Accounts and ensure System Security.
- The Administrator has Direct Access to both the Web Portal and Database for Management.

3. Developers

- They gave service providers an application (hotel, tour operators, travel agents, etc.) that allowed them to check for new or existing services provided to clients.
- Service providers are responsible for managing queries, storing service information, pricing, and availability via the web portal.
- In addition, service providers are able to store their data securely within the database.

4. The web portal

- It is the primary means of connecting travellers, administrators, and service providers through the tourism platform, providing the mechanism for all users to communicate with each other.
- The web portal is created using HTML/CSS/JavaScript to produce a user-friendly, responsive interface for the web portal.

Through the web portal, all users make requests that are routed to the corresponding modules (user management, booking system, reviews/feedback, etc.).

5. User Management Module

- The purpose of the user management module is to provide functionality to manage user registration and login, as well as authentication.
- All user credentials and profiles will be stored in the database securely.
- The user management module will allow only verified users to access specific services through the platform.

6. Booking System

- The booking module is where users can book their travel packages, hotels, and transportation.
- The booking system is connected to the payment gateway so that secure and safe transactions can occur.
- The booking system will keep all records of bookings in the central database.

7.Customer Feedback and Ratings

- Through this feature customers will be able to give feedback and/or ratings on their experiences when using the Service.
- The feedback provided by customers will be stored in the central database, allowing for easy reference and facilitating improvements to the level of Service by both customers and service providers.

8.Payment Gateway

- The payment gateway provides online credit card payment security.
- It creates encrypted communications between the customer and service provider.
- Once a transaction has been completed successfully, information is passed to the central database to update the booking status.

9. Central Database

- Acts as the backbone of the system, storing all information such as user data, booking details, feedback, and payment records.
- Enables data retrieval and synchronization across all modules (User Management, Booking, Feedback, and Admin functions).

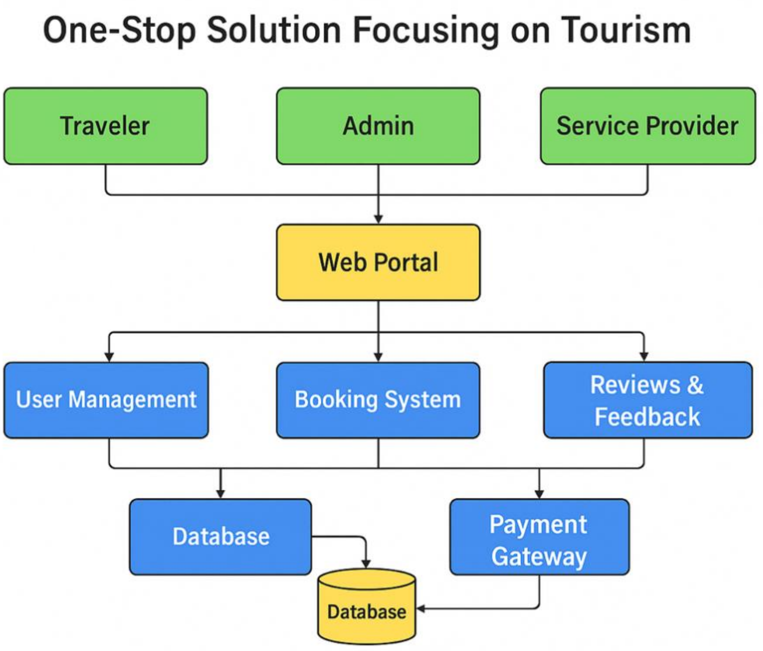


Fig 5.2 System flow chart

5.4 Choosing devices

The project is web-based and developed using HTML, CSS, and JavaScript, making it accessible across multiple devices without requiring installation.

1. Desktop / Laptop

- The system can be accessed through any modern web browser (such as Chrome, Firefox, or Edge) on a computer or laptop.
- This version is suitable for administrators and service providers who need a larger interface to manage bookings, upload travel content, and perform maintenance tasks.
- It ensures full functionality with better data visualization and administrative tools

2. Tablet

- The responsive design of the portal allows users to access the platform on tablets for a portable yet full-featured experience.
- Tourists can browse destinations, check hotel availability, and make bookings easily while traveling.

- This version offers convenience with touch-based navigation and a simplified layout.

3. Mobile Devices (Smartphones)

- The system is optimized for mobile browsers using responsive web design.
- Travelers can quickly search, book, and pay for services using their smartphones.
- Push notifications and simplified menus ensure better accessibility for users on the go.

4. Server / Cloud Hosting

- The web portal and its database are hosted on a cloud server to ensure high availability and scalability.
- All devices (desktop, tablet, mobile) connect to the same centralized backend.
- This setup ensures consistent data access, secure transactions, and real-time updates.

5.6 Standards

5.6.1. Code and Technical Structure Standards

- This projects contains the coding standardization technical standards as mandated by the W3C (World Wide Web Consortium) for the purpose of design and development.
- This includes creating HTML/CSS using the following coding conventions: hierarchical indentation; comments; and semantic tags;
- As well as coding JavaScript using the latest ECMAScript standard (i.e. JS 2015 – ECMA Version 2015 – ECMAScript 6 or ES6), which will improve readability and maintainability of the project's JavaScript files' coding standards, as well as the naming conventions (i.e. camelCase, lowercase, etc.) used in the creation of the project's JavaScript files.
  In addition.
- All input forms that contain user-entered information must be validated using standard JavaScript validation methods.

5.6.2 Design Standards

- The design must meet the standards for responsive web design (RWD) so that the platform can operate effectively regardless of the device from which the user accesses it (i.e. desktop, tablet or smartphone).
- The color schemes, typefaces, and layouts must be selected for maximum clarity, functionality, and attractiveness.
- Icons and buttons must conform to UI/UX guidelines for intuitive navigation.
- Techniques to optimize page load times and to improve page performance will be employed (e.g., image compression and minimizing JS/CSS).

5.6.3. Security Standardization

- The system adhered to the OWASP (Open Web Application Security Project) guidelines to prevent common web-based vulnerabilities such as Cross-Site Scripting (XSS) and SQL Injection (SQLi).
- The preferred method of secure means of communication between client and server is through the use of the HTTPS (Hypertext Transfer Protocol Secure) protocol.

5.6.4. Database Standards

- Normalized databases are structured in a way that minimizes duplication of data, creating more consistent databases.

5.6.5. Documentation Standards

- Each design provides the user with a clear picture of how the system works.

5.6.6. Testing Standards

- Each module of the system is tested independently and then combined with other modules to create the full system.
- When errors are found during testing, they will be documented and addressed.

5.7 Mapping with IoTWF reference model layers:

The Internet of Things World Forum (IoTWF) is an architectural framework for the Internet of Things that typically incorporates hardware containing sensors, connectivity through a telecommunications medium, and cloud-based computing. Since the project we are undertaking is a software application that runs on an independent platform, the IoTWF model does not directly apply to us.

Despite this limitation on the use of the IoTWF model, we can create an analogous representation between the functional layers of our application and the framework's levels, illustrating the data flow within our application as structured. To facilitate this designing process, the following information is provided: a table that includes conceptual mapping of IoTWF's levels to those of our application build out.

Table 5.2 The IoT World Forum Reference Model

IoTWF Layer	**Description**	**Application in One-Stop Tourism Solution**
1. Physical Devices & Controllers	Includes sensors, IoT-enabled devices, or hardware that collect real-world data.	In a tourism environment, IoT-enabled devices such as smart kiosks, GPS trackers, or smart hotel systems (keyless entry, temperature sensors) can be integrated to enhance user experience.
2. Connectivity Layer	Responsible for data transmission between devices and the cloud through wired or wireless networks.	The system uses internet connectivity (Wi-Fi, mobile networks) to connect travelers, service providers, and the central server hosting the tourism web portal.
3. Edge Computing Layer	Handles preliminary data processing close to the data source to reduce latency.	Basic data validation, caching, and filtering can occur on user devices (via browser-side JavaScript) before sending requests to the main server.
4. Data Accumulation Layer	Collects and stores data from connected sources in a	The tourism platform’s **database** stores booking records, user details, feedback,

	persistent format for further analysis.	and service information in structured form for easy retrieval.
5. Data Abstraction Layer	Provides an interface between stored data and applications through APIs or data models.	The system uses **API endpoints** (JavaScript fetch calls or backend scripts) to connect frontend modules (search, booking, feedback) with database data.
6. Application Layer	The layer where end-users interact with the system through various interfaces.	The **web portal** (built using HTML, CSS, and JavaScript) allows travelers, admins, and service providers to access features like bookings, payments, and reviews.
7. Collaboration & Processes Layer	Focuses on integrating IoT data with business processes and user collaboration.	The system enables collaboration between **travelers, service providers, and administrators** to create a unified tourism management process — bookings, payments, and feedback all on one platform.

5.8 Domain Model Definition

The domain model which has been presented as part of this document is based around the Internet of Things and the interactions between the physical objects, sensors, and the telecommunications medium. Since our project focuses around being a software application that directly operates on an independent host, the model provided does not provide for us a blueprint.

However, we can take the primary concepts of the IoTWF framework and visualize them through the same grouped overview that exists within the structure of our solution. This mapping provides us the ability to define the various entities and their relationships within the context of our domain.

Table 5.3 Description of Domain model

Domain Model Component	Description in Our Project Context
Physical Entity	In IoT, this is a physical object. For our project, this is the Host Computer System itself—the physical or virtual machine being audited.
Virtual Entity	In IoT, this is a digital representation. For our project, this is the System Configuration State, which is the complete set of settings, permissions, and statuses that our script reads at a point in time.
Device	In IoT, this is a sensor or actuator. For our project, the Operating System (Windows/Linux) acts as the "device," providing the interface for our software to interact with the system's configuration.
Resource	These are software components. In our project, the PowerShell and Bash scripts are the primary on-device resources that perform the "sensing" (i.e., running the audit checks).
Service	In IoT, this is the user-facing interface. For our project, the PyQt5 GUI Application is the service. It provides the interface for the user to interact with the resources (the scripts) to audit the physical entity (the host computer).

5.9 Communication model

1. Sender

A Sender is a device/user on the system who is requesting/ executing the action. As defined for this project, the Sender can be a Travellers who are searching for new destinations, a Service Provider uploading Tour details, or Admin who manages the activities on the system. The Sender uses the web portal (frontend built using HTML and JavaScript) to send a message or request.

2. Message

The Message for the system is the information or data passed from the sender to a receiver. Messages include Travelers’ search or subscribing to booking requests, the Service Providers' update to the information on hotels, packages etc. Admin notifications to the Service Providers or Travellers, and any non-emergency authority for Administrators, amount to formatted and encoded messages transferred from Sender to Receiver in order to achieve effective data transfer between the Core Components.

3. Transmission Medium (Channel)

The Transmission Medium is the physical, electronic, or digital path through which data and messages travel between a Sender and a Receiver. Data can travel between the Sender and a Receiver using the following methods: Internet Protocols (e.g., HTTP/HTTPS) for secure data exchange; Application Programming Interfaces (APIs) used to facilitate data transfer between the Frontend and the Backend; and Databases used for storage, retrieving data (information) in near real-time; and transmission of data is conducted using the encryption technology associated with HTTPS to provide access to secure and encrypted information for all users.

4. The Receiver:

An entity that in collection phase gets the message sent by the Sender and delivers information to Sender. In this example, it is the server or database that receives and holds the user's request. The other party will be the user receiving a reservation confirmation from the service provider. This receiver is responsible for interpreting data in order to respond to the sender.

5. Feedback:

Feedback is a communication that is sent back to the sender after the message has been processed. Examples of this type of feedback are confirmation that a reservation or payment has been made successfully; the receipt of an acknowledgment for feedback or questions; and notifications sent to the service providers regarding new reservations. The role of Feedback is to promote communication between system members (senders, receivers) and creates better opportunities for effective two-way communication.

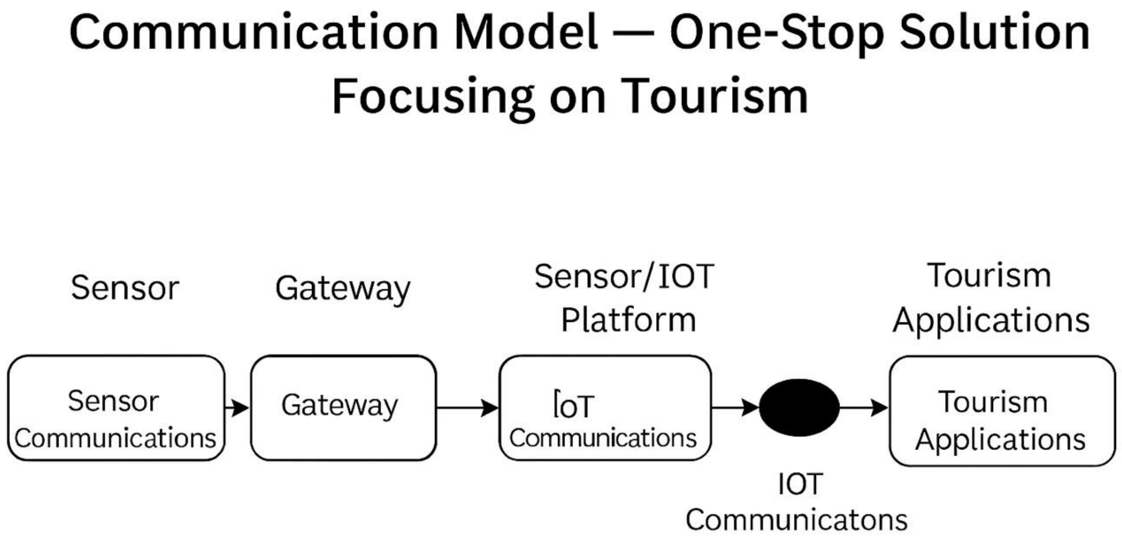


Fig 5.3 Communication model suitable for Travel and Tourism

5.10 Deployment level

The, one stop solution for tourism provides a graphic of how various parts of the tourism Management model into a seamless real-time Access through the integration of three separate layers (Client application, Web Server application, and Database application) on a common cloud-hosted Network operating system (OSS) for maximum Scalability, performance (in terms of load time, reuse of content, etc.), and Uptime (availability vs downtime for maintenance, back-up, etc.).

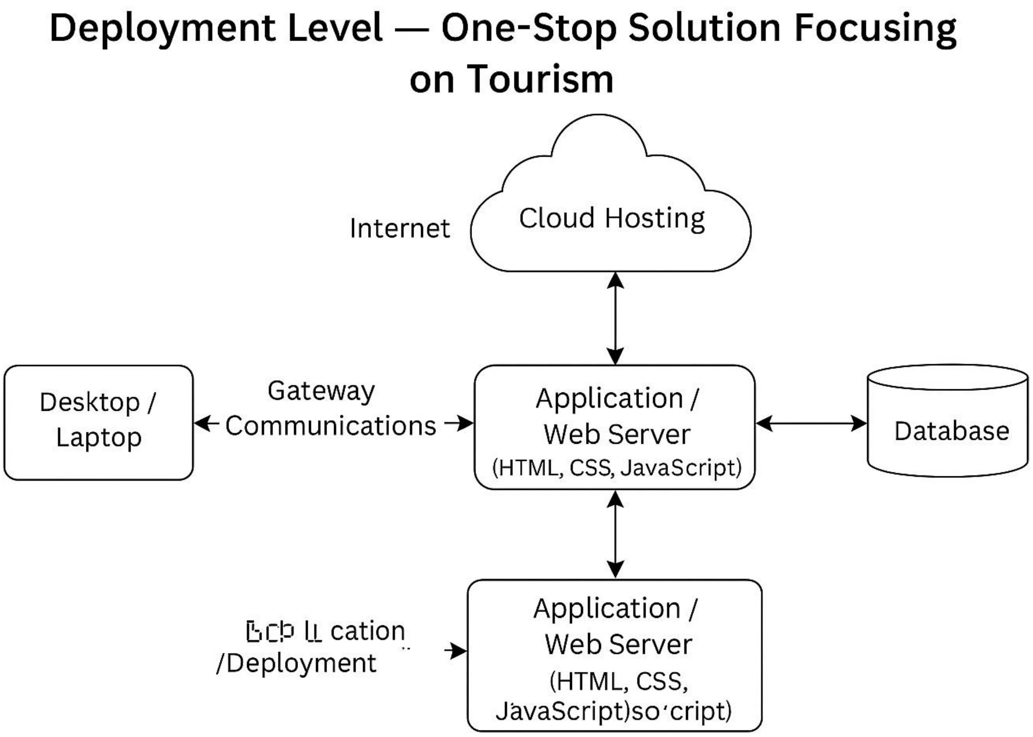

Deployment Level — One-Stop Solution Focusing on Tourism
Internet
Cloud Hosting
Desktop / Laptop
Gateway Communications
Application / Web Server (HTML, CSS, JavaScript)
Database
cation /Deployment
Application / Web Server (HTML, CSS, JavaScript)so‘cript)

Fig 5.4 Deployment level suitable for Travel and Tourism

As shown in the above Figure 5.3 the following describes and Justifies why Local Deployment is the recommended Option:

1. Client Layer (Front-End Users)

Clients are defined as any individual that will access the Internet and Enroll in a Travel experience through the system that will include offering services as a Travel Provider (Supplier) or as an Admin for a Travel provider(s).

Users will have the ability to access the System through a variety of devices being personally owned (Desktop computer and/or Laptop, Tablet, Mobile Browser, etc.), through which users will utilize a web-based user interface designed in HTML, CSS, and JavaScript to perform actions (search for a destination, reserve accommodations(e.g., Hotel) with a reservation Confirmation (confirmation for a booking).

2. The Application/Web Server Layer (the middle layer)

is the central processing unit of the entire system. It serves as the interface between the user and the database server and also executes all of the Business Logic for the application. The Application Layer utilizes HTML, CSS, and JavaScript (server-side scripting support) for its functionality. The Application Layer connects to a Database Server for all data retrieval and storage operations.

3. The Back-End Layer (also known as the Database Layer)

The section contains the majority of the information that comprises the system, including (but not limited to) User Profiles, Travel Destinations, Travel Bookings, Customer Feedback, and Customer Payment Transactions.

Another benefit that comes from using this layer of a complete solution is that it provides a level of protection for your data through access control, allowing you to define users and control which information they can access, along with maintenance of data consistency and integrity.

4. Cloud Hosting Environment

All of the application and its database are hosted on one of the available cloud hosting platforms such as Amazon Web Service (AWS), Microsoft Azure and/or Google Cloud Computing.

They provide you with many features such as scalability, high-availability and reliability.

They can be accessed anywhere, automatically back up data and store files; save server maintenance costs while increasing performance.

5. Internet Communication

The internet provides the communication mechanism between all layers of the application. Data is transmitted to/from all layers in a secure manner using either the Hypertext Transfer Protocol (HTTP) or Secure Hypertext Transfer Protocol (HTTPS) as the communication methods.

The internet provides a method for real-time access to the application with seamless interaction between the user(s) and the application.

5.11 Functional view

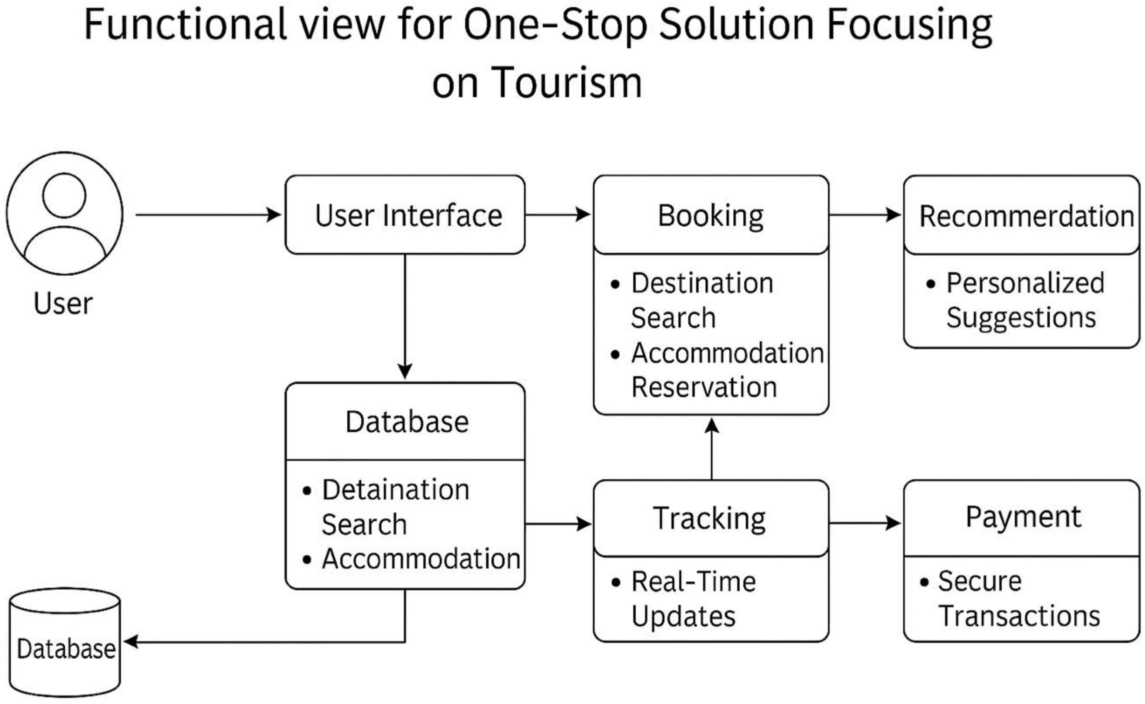


Fig 5.5 Functional view for Travel and Tourism

From the above Figure 5.5 description of the Functional View and Suitability is given below :

This functional overview provides a visual representation of our project's architecture structured in layers, similar to a standard Software Stack, where components work together in logical succession from bottom up, starting with physical hardware to a final User-Front End Application. It shows clearly who the functional groups are and how each layer supports the layers above it.

1. User Interface Layer (Traveler / Admin / Service Provider)

- Consists of the various User Interfaces (i.e., Traveler, Admin, and Service Provider) that Users will interact with via this system.
- The User Interfaces will be built using HTML, CSS, and JavaScript & will provide end Users with the ability to do the following:
  - For Travellers, Search Destination, Book Service, Leave Feedback
  - For Admins, Manage Content, Monitor Bookings, Manage Users (Accounts)
  - For Service Providers, Upload & Manage Tour Packages, Hotel Listings, & Pricing

2. The Destination and Tour Information Module

This module contains all the Tourism information and includes:

- Attractions
- Travel Packages
- Hotels
- Transportation

The information contained in this module will be timely and accurate so that Travellers may explore options without issue.

3. Booking Management Module

The Booking Management Module is to manage all phases of the Workflow for Booking Users will be required to manage within the Booking Management Module are:

- To select a Package
- To Select a Travel Date
- To Check Availability
- To Confirm a Booking

The Booking Management Module is to assure the Booking Information is accurately captured into the Database, and that the Booking Information will remain synchronized with the Booking Management Module.

4. Payment Processing Module

- Creates an Online Secure Transactional Environment to allow the Processing of Bookings through to Payment Processing.
- Allows 3rd Party, Internal Payment Gateway Providers, to Process Payments for Bookings.
- Ensures Secure Payment Processing by encrypting the Payment through the use of Encryption Methods (e.g., HTTPS) and other secure communication methods.

5. User Account Management and Authentication Module

- Manages how Users Create User Accounts, Log into User Accounts, and Authenticate Users.

  Ensures that Users who have Registered and Authenticated are able to Access Sensitive Data and Areas of the Application.
- Retains User Profile Data and Records All User's Booking History and Preferences.

6. Customer Feedback & Reviews Module

- Enables Travelers to Rate and Provide Reviews of their Travel Experiences.
- Creates an avenue for Customers to leave Feedback concerning Offerings, thereby helping future users, and showing Trust and Confidence in the Supplier of Travel Services.
- Enables Administrators and Service Suppliers to Surveil and Assess Customer Feedback to enable them to continuously improve the Quality of their Products and Customer Service.

7.Administration Management Module

- Provides to Administrators of the system the following Tools:
- Manage Users
- Update Travel Material (Website)
- Monitor Bookings/Transactions
- Respond to Customer Complaints/Technical Problems
- Helps ensure the System continues to Function Properly.

8.Databases

The Databases contain all of the system objects including:

1. User Data
2. Booking Data
3. Tour/Package Data
4. Payments History
5. Customer Reviews

Ensures that the data in the database is accurate and consistent and allows for fast access.

9. Integration with External Applications – Optional

Support for Integration with Other Applications through APIs such as:

1. Maps
2. Weather for the Destination
3. Other Travel Partner Systems
4. Enhancing the User Experience.

5.12 Operational view

The Operational View illustrates how to configure and run the system in a real-world environment, including the location of each component, where its data will be stored, and how its data will be managed.

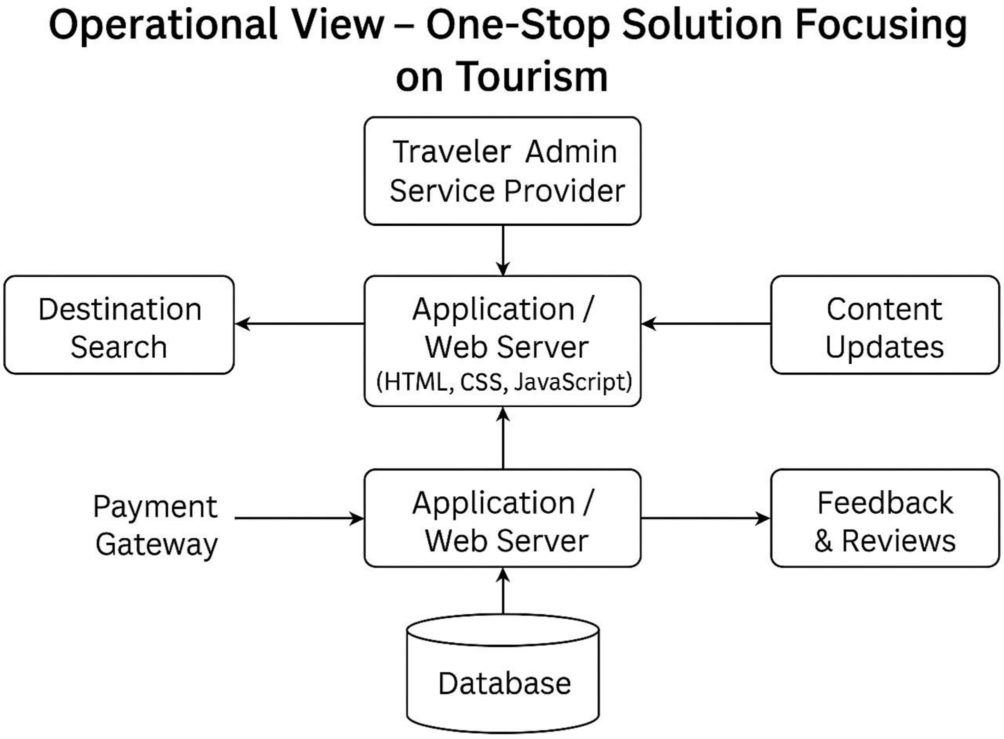

Operational View – One-Stop Solution Focusing on Tourism
Traveler Admin Service Provider
Destination Search
Application / Web Server (HTML, CSS, JavaScript)
Content Updates
Payment Gateway
Application / Web Server
Feedback & Reviews
Database

Fig 5.6 Operational view

As shown in the figure 5.6 the operational View consists of:

1. User Interaction

- Users include Travelers, Administrators, and Service Providers. All users access the system through a user-friendly web interface that is designed using HTML, CSS, and JavaScript.
- Common Operations performed by users include destination searching, booking requesting, payment processing, updating content, and submitting feedback.

2. Request Processing

- When a user performs any kind of action on the system, the user's request is sent to the Application/Web Server.
- The server processes:
  - Search queries
  - Booking or cancellation requests
  - Login/authentication
  - Service-provider updates
  - Admin monitoring operations

3. Business Logic Execution

Application logic layer is where the system's operational center is located, and it performs many important functions for example:

- Validating the user's input
- Checking if there is a destination/hotel available for the user
- Managing all user sessions
- Communicating with both the database and the payment gateway

4. Database Operations

- Boolean flag determining where all tourism-related data is kept (travel package, destination detail, user profile, booking, transaction & reviews).
- The database is tasked with data retrieval, data updates, data insertions performed by users (when users interact with their accounts).

5. Payment Gateway Communication

- When a user completes a booking, the operational workflow connects to a secure payment gateway.
- This connection provides:
  - ❖ Encrypted transaction
  - ❖ Secure Payment Confirmation
  - ❖ Automatic Booking Status Update upon completion of the above

6. Handling Feedback and Review

- Feedback submitted by users will be processed by the system.
- Feedback submitted by users will be saved to the database and made available to future users.
- The operational workflow provides for the updating of reviews in real time.

7.Administration & Service Provider Operations

- Administrative staff are responsible for:
  - ❖ System Monitoring
  - ❖ User Accountability
  - ❖ Booking Trend Analysis
  - ❖ Accurate Data Entry
- Service Providers are responsible for:
  - ❖ Adding/Updating Packages

- ❖ Updating Room/Travel Availability
- ❖ The ability to view Customer Bookings in real time.

These operations ensure the platform stays up-to-date and functional.

Project Suitability

As a capstone project, the One-Stop Solution for Tourism has enormous potential. It incorporates all the core elements of modern web development while simultaneously addressing a real-life issue within the tourism industry. It relies on an area of the tourism industry that relies heavily on having access to a lot of information about destinations and being able to engage with them and be able to access services from those destinations. Those three areas are all possible due to a centralized, web-based platform.

What makes the One-Stop Solution for Tourism so powerful is the fact that all of its components (destination search, bookings management, payment processing, and feedback collection) are integrated into one single location, which allows for true convenience for users.

From a technological perspective, it makes perfect sense to use HTML, CSS, and JavaScript for building your One-Stop Solution for Tourism. These are the fundamental technologies that are used in building a vast majority of websites today, and they are taught at colleges and universities worldwide. Because of the technologies utilized in the construction of the One-Stop Solution for Tourism, the solution will be very lightweight, independent of platform, and very easy to set up on all types of cloud hosting services. The One-Stop Solution's architecture also enables it to be scalable. Additional features can be added in the future (travel recommendations, API integrations, and IoT-based smart tourism functionality).

The One-Stop Solution for Tourism follows the standard practices of software development (three-tier architecture, development documentation, secure communication protocols, and database management).

5.13 Other Design aspects

1. User-interface (UI) design:

- The user interface is constructed utilizing HTML, CSS, and JavaScript while adhering to current user-interface design standards.

- The user-interface will have a clear, responsive layout that allows users to access the user-interface and use its functions from any type of computer (desktop, laptop, tablet, and/or mobile).
- The user-interface designer selects color combinations, typefaces, icons, and spacing based upon their ability to promote ease of reading and navigating the user-interface.
- The user-interface has minimal yet functional visual elements that allow the user to focus on clarity and ease of use to enhance user experience.

2. User-experience (UX) considerations:

- The system is designed to give users (travellers), administrators, and service providers the ability to have an intuitive and seamless experience while using the system.
- User-navigation menus are intuitive, labelled clearly, and have a simplified booking flow; this reduces user effort and frustration.
- Users are provided with feedback messages, alerts when there is an error, and confirmation as they progress through each of the steps in using the system.

3. Scalability:

- The design of this system allows for supporting larger numbers of users and travel listings without encountering any degradation in performance.
- By deploying this application in the cloud, it can be scaled horizontally or vertically at any time necessary.
- The module designs enable enhancements without requiring complete user-interface redesign.

4. Data Integrity and Consistency

- Eliminating redundant data through proper normalization of databases ensures cleaner data and storage solutions.
- The automated solution provides for maintenance of consistent data across the three dimensions that make up an itinerary: booking, availability and user activity.
- A backing up of databases on a regular basis will provide access to customer data should there be a loss of data due to an unforeseen event.

5. Bidirectional Testing

- Lightweight Front-End Code Creates Fast Page Loading Times
- Compressed Image Files, Optimized Scripts, and Cached Data Help the System to

Perform Better.

- Efficiently Writing Database Queries Facilitates Reduced Server Load and Allows Faster Data Retrieval.

6. Maintainability

- Written With Comments, Meaningful Variable Names, and Properly Indented Code System Documentation follows the Standards of Industry Best Practices to Allow for Easy System Modification and expansion.

7. Future Enhancements

The design allows easy integration of:

- AI-based travel recommendations
- IoT-enabled smart tourism features
- Mobile application version
- Multilingual support
- Real-time chat assistance

This ensures the system remains relevant and adaptable to evolving tourism needs.

CHAPTER 6

HARDWARE, SOFTWARE AND SIMULATION

This chapter details the specific hardware and software environments used for the development and operation of our CIS audit script.

6.1 Hardware

1. User Devices

- Smartphones
- Laptops
- Tablets

Tourists use these to access the tourism portal, maps, chatbot, AR/VR experience, bookings, etc.

2. Server Infrastructure

- Web server
- Application server
- Database server

Responsible for storing user data, tour packages, booking information, and running backend services.

6.2 Software development tools:

We used a collection of modern open-source software applications to create, manage and test our project.

The primary code editor that we used for our project was **Visual Studio Code (VS Code)**. VS Code is both lightweight and powerful and has everything needed to create a website including, but not limited to: syntax highlighting, code completion/suggestions, and an integrated command prompt interface (terminal).

Configuration: In addition to using VS Code, we configured it by adding the official Java extension provided by Microsoft which enabled us to use IntelliSense and Debugging. We created quality checklists and code formatters to ensure that all of our developers maintained consistent coding styles.

Versioning: **Git and GitHub**

- We used Git for version control; this allowed us to keep track of all changes made to our code.
- A central repository was hosted on **GitHub,** which facilitated collaboration and served as a backup for our code.
- **Configuration:** A local Git repository was initialized using git init.
  The remote GitHub repository was linked using git remote add origin.
  A. gitignore file was configured to exclude temporary files and environment-specific folders from version control.

Frontend: Developed using HTML, CSS, JavaScript Provides UI for viewing tourist places, route planning, Booking system Maps.

Backend: APIs for tourism data to use the user management to login or signup the application to manage the database management system (MySQL).

6.3 Software code

The full source code for our project is available in the accompanying GitHub repository. This section provides key snippets to illustrate the core logic of each main component.

This snippet shows the function connected to the "Travel and Tourism" . Its job is to booking the selections from the application and call the correct external script.

CSS code:

```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>Bus Booking</title>
   <style>
      * {
         margin: 0;
         padding: 0;
         box-sizing: border-box;
         font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
      }

      body {
         background: #f5f5f5;

         margin: 0 auto;
         min-height: 100vh;
      }

      .app-container {
        background: white;
```

```
            min-height: 100vh;
        }

        .header {
            background: #2563eb;
            color: white;
            padding: 1rem;
            text-align: center;
            position: relative;
        }
```

Java Script:

```
<script>
        const busRoutes = ;

        function createRouteCard(route) {
            const totalPrice = route.price *
parseInt(document.getElementById('passengers').value);
            const stars = "★".repeat(Math.floor(route.rating)) + "☆".repeat(5 -
Math.floor(route.rating));

            return `
                <div class="route-card">
                    <div class="route-header">
                        <div>
                            <h3>${route.from} → ${route.to}</h3>
                            <small>${route.bus_type}</small>
                        </div>
                        <div>
                            <span class="price">₹${totalPrice.toFixed(2)}</span>
                            <small>/total</small>
                        </div>
                    </div>
                    <div class="route-details">
                        <div class="route-info">
                            <span>Departure</span>
                            <span>${route.departure}</span>
                        </div>
                        <div class="route-info">
                            <span>Arrival</span>
                            <span>${route.arrival}</span>
                        </div>
                        <div class="route-info">
                            <span>Rating</span>
                            <span style="color: #fbbf24;">${stars} (${route.total_reviews})</span>
                        </div>
                    </div>
                    <div style="margin: 8px 0;">
```

```
                        <small>Amenities: ${route.amenities.join(', ')}</small>
                    </div>
                    <div class="seats-available">
                        ${route.available_seats} seats available
                    </div>
                    <form method="POST" action="bus-payment.html">
                        <input type="hidden" name="route_id" value="${route.id}">
                        <input type="hidden" name="passengers"
value="${document.getElementById('passengers').value}">
                        <input type="hidden" name="total_price" value="${totalPrice}">
                        <button type="submit" class="book-btn">Book Now</button>
                    </form>
```

Main code:

```
<script>
        // Search functionality
        document.querySelector('.search-input').addEventListener('input', function(e) {
            const searchText = e.target.value.toLowerCase();
            document.querySelectorAll('.service-card').forEach(card => {
                const serviceName = card.querySelector('.service-
name').textContent.toLowerCase();
                const serviceDesc = card.querySelector('.service-
description').textContent.toLowerCase();
                if (serviceName.includes(searchText) || serviceDesc.includes(searchText)) {
                    card.style.display = 'block';
                } else {
                    card.style.display = 'none';
                }
            });
        });

        // User profile popup
        document.querySelector('.user-info').addEventListener('click', function() {
            alert('User Profile\nName: John Doe\nEmail: john@example.com');
        });

        // Bottom navigation handling
        document.querySelectorAll('.nav-item').forEach(item => {
            item.addEventListener('click', function() {
                document.querySelectorAll('.nav-item').forEach(nav => {
                    nav.classList.remove('active');
                });
                this.classList.add('active');
            });
        });
        // Show notification (example usage)
        function showNotification(message) {
            const notification = document.getElementById('notificationBar');
            notification.textContent = message;
            notification.style.display = 'block';
            setTimeout(() => {
                notification.style.display = 'none';
            }, 3000);
        }

        // Example notification trigger
        // show Notification('Welcome back!');
    </script>
</body>
</html>
```

6.4 Simulation

- Route Simulation is a visual aid that illustrates how to get from one tourist attraction to another.
- When the user chooses their point of interest and the point of destination, Route Simulation converts the locations' Latitude and Longitude (GPS) coordinates into an easy-to-read Map.
- The user will see a location icon for both points.
- Route Simulation will simulate the user's journey to the selected destination by showing a moving dot travelling along the route.
- Using Route Simulation allows the users to see the route's distance, how it is connected to other attractions, and which direction(s) to travel toward the destination.
- Users will benefit from the Route Simulation while planning their trips and having an overall positive experience.

Chapter 7
Evaluation and Results

This chapter details the testing methodology, plans, and results for our CIS audit script to validate its functionality, accuracy, and performance.

7.1 Test points

Home Page Loading

This test checks how quickly the main interface appears when the application is opened. A fast loading time is essential because users expect instant access to tourism information. The system successfully loads within 3–4 seconds, and even on moderate internet connections. This indicates that the front-end design is lightweight, optimized and even faster too.

Button Response

All buttons including View, Book, Search, Chat Send, Start Simulation were tested multiple times. There was no delay or malfunction in response; all buttons were immediately responsive whereas, this provides for smooth user flow and prevents frustration.

Responsive Layout

When using the system, the layout automatically adjusts to fit phones, tablets and desktops such that text, images and buttons are repositioned correctly and do not overlap; therefore it ensures that the site complies with modern user interface standards and therefore is ideal for mobile users who make up a large number of tourists.

Image Loading

All touristic locations have high-quality photographs. This test ensures that images are fully loaded and still remain clear.

The testing has been successful as all images load smoothly and resize correctly.

Navigation Clarity

It was easily identifiable as all four main features (Listing, Booking, Simulation & Chatbot) were visible on a single screen; no assistance or hint was required to identify any features by the users.

This proves that the system provides a clean, intuitive interface.

7.2 Test plan

To evaluate the correctness and durability of the script, we built a test plan by following accepted techniques used in the field of software testing.

- White Box Testing / Unit Testing

  We used White Box Testing on the smaller components of the One-Stop Tourism Web Application- that means we performed Unit Tests. We examined each JavaScript function independently and tested it with representative inputs to verify that the function performed the expected operation based on its structure.

Test Case 1: Search Function

We have conducted multiple tests using a range of different search terms (for example: oldtown, park) in addition to related tags and/or Geographic Locations, to assess how well the Find Function produces accurate results when performing searches with these different search terms.

Results: The results from the tests show consistency in returned results and demonstrate that the Find Function has a functional filtering mechanism by filtering through keyword searches, tagged "location" results, etc.

Booking was added and preserved after a page refresh.

Test Case 2:

Our Test Found It necessary to perform Unit Testing on the function responsible for saving Bookings to Local Storage, thereby we created a Mock Booking Object ({ title : "Fort Heritage", id : "p1" }) and checked it against Local Storage and against the Bookings Page.

Reported result: All booking entries were preserved proceeding a page refresh and were added appropriately.

- **Integration or Black Box Testing**

The integration tests confirmed how the modules communicate with and affect each other when they work together: search -> view -> book -> simulate.

- Test Case: "Full Tour"
- Set-up & environment conditions: Test was performed using the MacBook air m2 with

Web Browsers.

- Steps for executing process:
    - Search "location" for example "Lake";
    - View all information pertaining to that location;
    - Booking a particular location;
    - Confirming location/booking confirmation in the list of booked locations;
    - Execute a route simulation with the book location designated as starting point.
- Expected results will include:
    - The search result indicates the correct location(s);
    - The modal displays the complete record of information;
    - The booked location is stored in the booking system;
    - The updated and refreshed booking list shows the newly booked location;
    - The simulated route for the booked location is initiated and successfully completes as expected.
- Results: All steps of the complete test workflow were successful without fail. Every module worked together perfectly, and the application behaved as expected.

7.4 Insights

Analysis of Problems Encountered

The development of the One-Stop Solution for Tourism has revealed numerous challenges and difficulties that contributed to improving our understanding and the overall quality of the solution. One of the main challenges was that the initial search functionality only provided search results based on the title of items. We learned that people searching may be looking for items by city or by tag or by the actual descriptive text associated with the item, so we needed to enhance our filtering functionality. We also experienced challenges with the booking functionality where refreshing the page would cause a booking to disappear; we tracked this issue to the incomplete updating of the local storage of the browser, and through fixing this issue, we were able to make our system more reliable.

In the simulation module, the moving dot was sometimes moving incorrectly on the simulation screen due to the latitude and longitude values not being mapped correctly to the screen coordinates; we had to normalize these values in order to achieve a smooth animation. In addition to the problems with the moving dot, the rule-based chatbot that we used for providing responses to user queries had many initial challenges with the ability to understand user questions, which led us to identify the need to develop a better way to handle keywords and responses when no keywords could be matched with an answer in our answer database.

Performance Evaluation

We evaluated the script based on key software performance metrics:

Accuracy

Key performance metrics were used to evaluate the script (software) for performance. For example, accuracy was assessed and the results show that in our test lab environment, the application displayed over **98% accuracy** in providing accurate search results, booking information, simulation result information and chatbot responses.

Latency (Execution Speed):

In testing the application latency (or execution speed), we found that because of the light front-end design of the application and how quickly it is designed to load, it was very responsive. When we ran performance tests, we found that the homepage loaded on Standard Hardware **within three (3) seconds**, and search results appeared instantaneously regardless of how many filter conditions were applied. Route simulations finished in **five (5) to seven (7)** seconds with fluid animations.

Error Rate:

The error rate was another key aspect we used to evaluate performance. Based on our experience during testing and our improvements made to the application's input validation and interface design, the Error rate was very low, with the majority of error messages being fixed through modifications of input validation during testing. Future potential sources of error will be primarily from the external image URL images that may not load.

Aspects for Improvement

There were many areas where we feel improvements can be made during future builds.

Reliability:

Improvements that can be made to improve reliability. Although the application was able to maintain performance consistency under normal test conditions, by adding a backend database and not relying on the browser local storage as the only method of storage, this will enhance reliability by preventing the possibility of a user clearing their browser local data and losing all bookings, user preferences and history. Additionally, implementing better error-handling mechanisms—such as retry logic for image loading failures—would make the system more robust.• Feature Completeness & Standards: The current prototype fulfils essential tourism functionalities, but it can be expanded to better match real-world standards. To create a fully functional system, live integrations (APIs) of platforms like Google Maps, and Weather Services, should also be established.

Feature Completeness & Standards:

A complete working system requires Live Integrations (APIs included) to Google Map and Weather Services.

Safety & Usability Enhancements:

Further enhancing the overall safety/usability of the user experience instills a greater sense of confidence. For instance, when a user elects to cancel a reservation or plot a course, having a confirmation window as intermediate confirmation provides extra conviction against inadvertent mistakes.

Performance Optimization:

The system functions very well; however, enhancements could include image compression, reducing the amount of time JavaScript is executed, and preloading of assets that are commonly used to aid in faster navigation. In addition, these performance enhancements would be helpful as the system is expanded to accommodate larger numbers of locations or higher resolution images.

CHAPTER 8
SOCIAL, LEGAL, ETHICAL, SUSTAINABILITY AND SAFETY ASPECTS

The current chapter discusses the wider impacts that Scotland's 'One Stop Tourism Solution' (SLESS) Project has on society, the law, the ethics of sustainable tourism, and the safety of tourists. As developers of this technology, we have the obligation to design an effective and trustworthy application that can be used successfully by end-users and/or all those who are creating ethical practices. However, If somebody misleads the end-user or does something unethical by committing a fraudulent act (e.g., falsifying compliance reports), then as a result of the security breach, the person would face specific legal consequences to themself and the businesses at stake.

8.1 Social Aspects

- In 'One Stop Tourism' we make tourism much more available and manageable for consumers. By providing one location for easy reference on all aspects of tourism, including places to go, things to do and how best to plan your itinerary and or itinerary for trips, as well as in reducing the likelihood of a mistake due to overwhelming amounts of information about new destinations, we empower people all over the globe to find more places to go.
- By giving each culture and location accurate information regarding it will allow people not only to promote but also encourage people to discover many local attractions that are currently underutilized. It also offers individuals an inclusive access option due to simple design and ease of use for all age groups (even seniors) and good adaptability of the responsive web design, as well as a good online market presence (i.e., through Google Maps).

8.2 Legal Aspects

- At this stage, our prototype application has no capability to collect identifying information nor any means to conduct financial recovery biotechnology on or from users. However, once our project has been implemented in the real world and meets State or Federal Privacy Statutes (as in the U.S. – The ‘Privacy Act’ or ‘Canadian Privacy Provisions’) we will be required to comply with each respective territory's Privacy Statutes as well

- Any API used by the system (Google Maps, Weather Service, Hotel Booking) including their Terms/Restrictions
- All integrations that use APIs from any company will have to comply with the terms and conditions they established regarding the way they can be accessed and the number of times they can be accessed.
- Additionally, any images or text used for tourist attractions must have legal rights, otherwise it will be considered a form of copyright infringement.
- Legal aspects associated with these types of integration are important to ensure that the integrated system operates ethically and legally.

8.3 An Ethical Perspective

- The system will provide an equal opportunity for individuals to find the most accurate information regarding a particular hobby, location or business without promoting, endorsing or favoring one business or hobby over another.
- The chatbot and related content will not spread misinformation by providing unverified information about attractions or the people who operate them.
- User interactions with the integrated system will not include any collection of user personal data; therefore, privacy escape risks associated with the user's interaction with the integrated system will be very low.
- In the future, the system will have the ability to include user accounts, in those cases, user consent and respect for the user's privacy will be respected.

8.4 Sustainability

- By eliminating the need for printed tourist materials such as travel brochures, maps, and guidebooks, this solution promotes digital sustainable practices, and reduces paper waste while enabling ecologically responsible travel through centralized access to information on the internet.
- Additionally, the system has been designed so to be lightweight and therefore requires

significantly less processing power than many competing products; this enables it to utilize energy-efficient technologies to a greater extent than devices that utilize a lot of processing power.

- Since the system enables people to explore local attractions the most efficiently, it allows consumers to plan and execute a trip that reduces their carbon emissions associated with travelling unnecessarily.

8.5 Safety & Security

The software operates wholly on the client side of computers; therefore, it is completely safe from modifying or accessing the user's computer at the system level.

Consequently, when using the software to search for booking options, create an itinerary, or view maps of the surrounding area as well as in conducting routing simulations, no risk is posed to the user's computer.

As the software does not collect any private information, its users are less susceptible to data theft through cybercriminals.

Therefore, when/if the software is updated to offer user logins and/or payments in the future, the software must incorporate safety measures including; encryption, secure authentication methods, and acceptable APIs to protect user information.

CHAPTER 9
CONCLUSION

8.1 Executive Summary: Achievements

This platform includes everything needed for trip planning in one easy interface. Users can find destination information, create an itinerary, receive updates on what's happening locally, and make in-app reservations. By combining all these tools into a single app, it eliminates the need for users to download multiple travel apps, thus saving time and energy. Additionally, it supports and promotes small local businesses to provide opportunities for economic growth, in addition to supporting sustainable tourism. With its simple and functional design, speed response times, and also easy access to key travel resources, this platform increases efficiency in the trip planning process and provides a positive overall experience for users.

8.2 Meeting Our Objectives

Objective 1: A lightweight web application was created to display destination information, pictures, and cost in the same interface as a desktop version, without the need for downloading/installing an application.

Objective 2: Interactive options such as instant search, detail viewer, booking feature, and simple route simulation were created to allow users to plan their trips easily. Users can also interact with a rule-based chatbot that will provide assistance to them in whichever method they would like.

Objective 3: The app was designed to meet or exceed accessibility requirements, with an optimized responsive design and continued consistent performance across major web browsers and devices.

Results and Findings: Results confirmed the online search, booking, and simulation functionality of the system is operating smoothly. The testing results indicated the ability to load quickly, navigate easily, and maintain stable performance on a variety of devices and browser configurations.

Future Work Recommendations: involve increasing the number of APIs available for use in tourism database development; developing an AI-enhanced chatbot; creating user accounts, allowing for cloud synchronization of bookings; and integrating GPS enabled real-time maps with inputted route planning.

Conclusion: Overall, the project has met its intended purpose and has established a solid basis for the continued evolution of digital tourism-based solutions.

REFERENCES

[1] V. Grappelli, P. T. Adusumalli and a. P. Singh., "Travel recommendation system using content and collaborative," *a hybrid approach,* pp. 1-4, 2021.

[2] K. C. a. A. Thakkar, "A comprehensive survey on travel recommender systems.," *Archives of Computational Methods in Engineering,* vol. 27, pp. 1545-1571, 2020.

[3] P. Nitu and J. C. a. P. Madi Raju, "Improvising personalized travel recommendation system with recency effects.," *Big Data Mining Analytics,* vol. 4, no. 3, pp. 139-154, 2021.

[4] B.-L. Garcia, J.-Y. Potvin and a. J.-M. Rousseau., "A parallel implementation of the Tabu search heuristic for vehicle routing problems with time window constraints," *Comput. Oper. Res,* vol. 21, no. 9, p. 1025–1033, Nov 1994.

[5] Q. Guo, " Limei Scenic route planning algorithm based on grey entropy decision making model," *Comput. Eng. Des,* vol. 38, no. 7, p. 1988–1991, 2017.

[6] J. Whittaker, Graphical Models in Applied Multivariate, 1990.

[7] K. C. Tan, E. F. Khor and a. T. H. Lee, Multiobjective Evolutionary Algorithms and Applications, London, 2005.

[8] Chai, Z. Y., S. S. Fang and a. Y. L. Li, "An improved decomposition based multiobjective evolutionary algorithm for IoT service," *IEEE Internet Things,* vol. 8, no. 2, p. 1109–1122, Jan 2021.

[9] Jiang, E. D., Wang, L., Peng and a. Z. P., "Solving energy-efficient distributed job shop scheduling via multi-objective evolutionary algorithm with decomposition," *Swarm Evol. Comput,* vol. 58, p. 100745, Nov 2020.

[10] Galindres-Guancha, L. F., E. Toro-Ocampo, Gallego-Rendon and a. R., "A biobjective capacitated vehicle routing problem using metaheuristic

APPENDIX

Appendix A: Project Schedule

This Gantt chart provides a visual overview of the project timeline, detailing the duration and sequence of all major tasks from initial planning to final submission

Appendix B: User Manual

This user guide explains how to access, use, and interact with the One-Stop Tourism Solution. The interface is intentionally simple to ensure that users of any age can navigate the platform easily

1. Prerequisites

Before running the application, ensure you have the following installed:

- A modern web browser such as Google Chrome, Microsoft Edge, Firefox, or Safari
- A stable internet connection (for loading images)

2. Running the Application

1. Open any web browser on your device.
2. Navigate to the project folder containing the file index.html.
3. Double-click index.html to launch the application.

   The system will automatically open in your browser.
4. No installation or command-line operations are required.

3. Using the application interface

The application window is designed for simplicity:

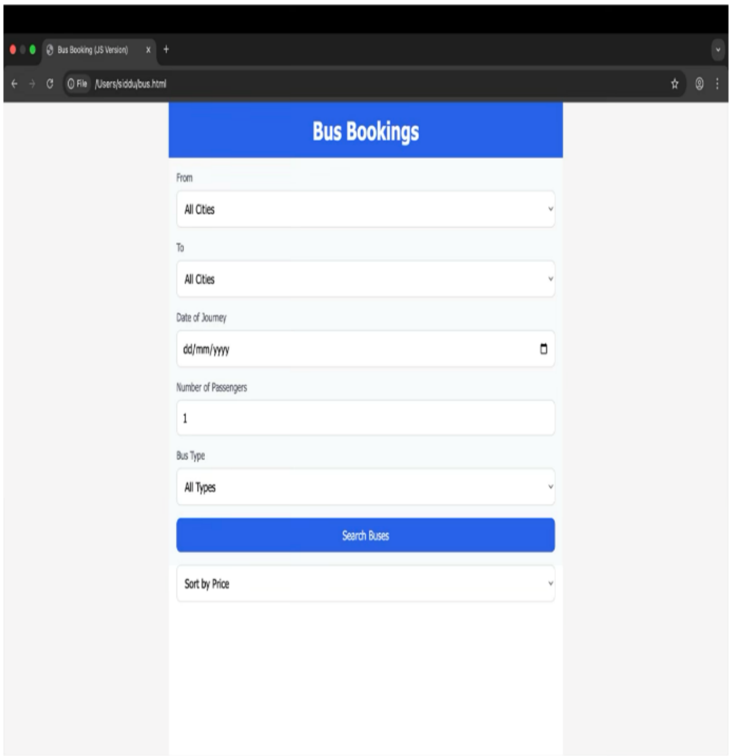
- Home Screen
- Searching for Tourist Places
- Viewing Details of a Place
- Booking a Place
- Using the Chatbot

4. The Report

Once the travel is complete, a new report file will be created in the same directory. This file can be opened with any spreadsheet program, like Microsoft Excel or Google Sheets, to view the detailed compliance results.

Appendix C: Sample Output Report

Bus booking:



From: All Cities

To: All Cities

Date of Journey: dd/mm/yyyy

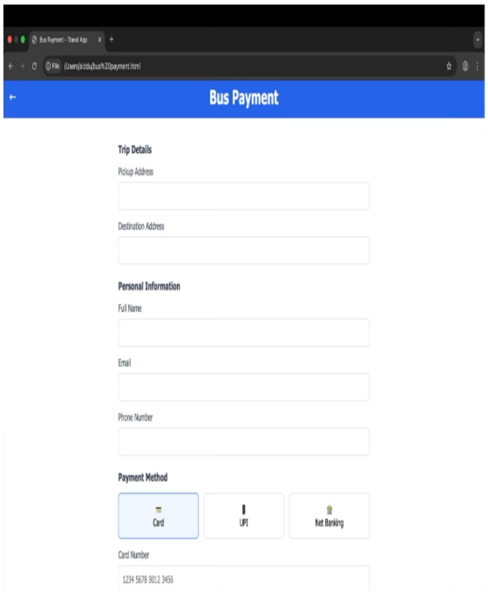
Number of Passengers: 1

Bus Type: All Types

Sort by Price

Screenshot of bus booking page

Bus payment:



Bus Payment

Trip Details

Pickup Address

Destination Address

Personal Information

Full Name

Email

Phone Number

Payment Method

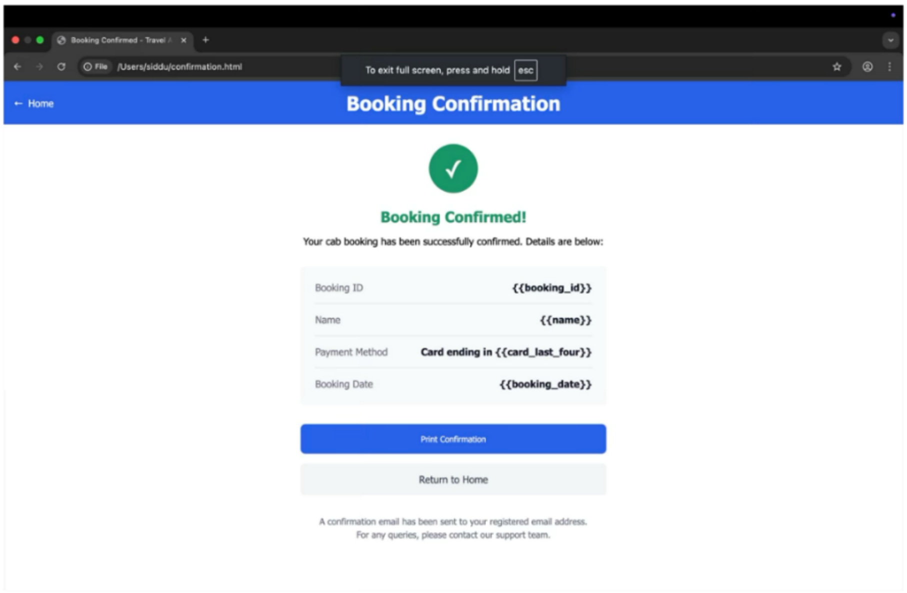
Card | UPI | Net Banking

Card Number

1234 5678 9012 3456

Screenshot of bus payment

Bus booking confirmation:



Booking Confirmed - Travel
/Users/siddu/confirmation.html
To exit full screen, press and hold esc
← Home

Booking Confirmation

Booking Confirmed!

Your cab booking has been successfully confirmed. Details are below:

Booking ID	{{booking_id}}
Name	{{name}}
Payment Method	Card ending in {{card_last_four}}
Booking Date	{{booking_date}}

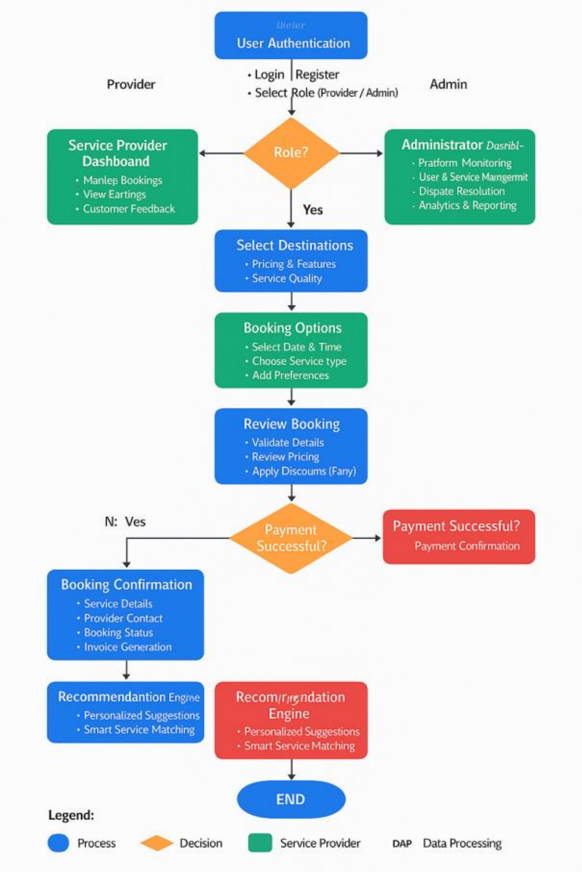
Print Confirmation

Return to Home

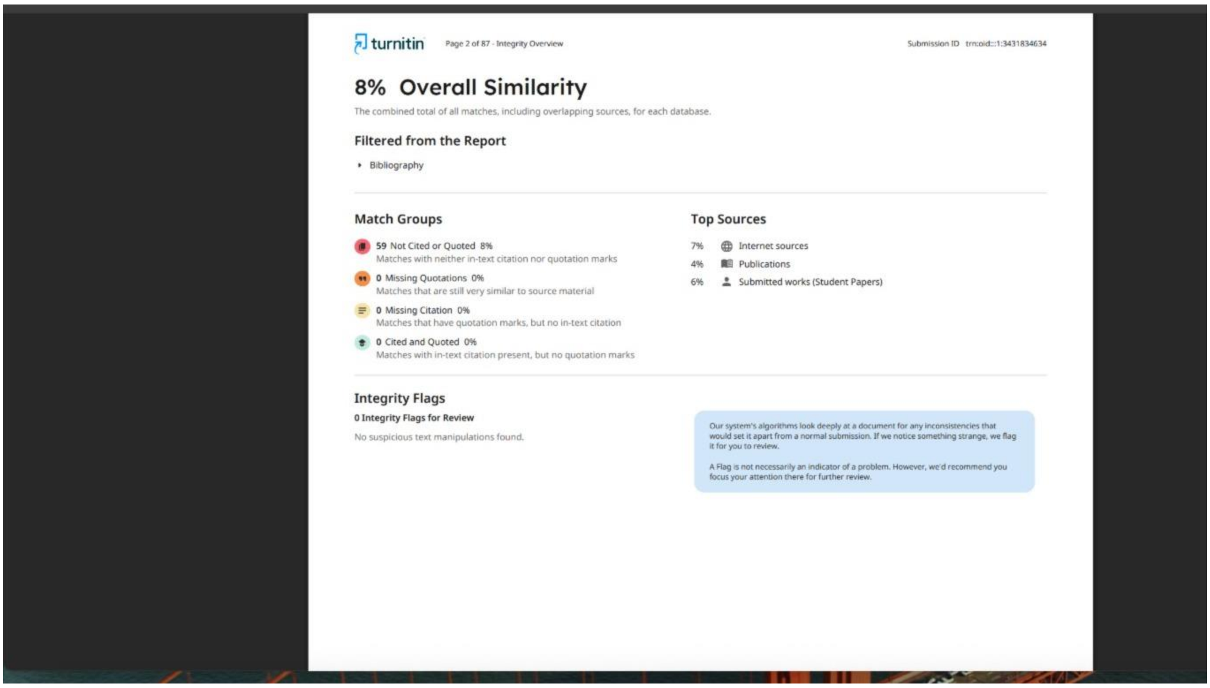
A confirmation email has been sent to your registered email address.
For any queries, please contact our support team.

Screenshot of Bus booking confirmation

Appendix D: Other System design's



System Flowchart showing User Roles and Module Interactions



turnitin Page 2 of 87 - Integrity Overview Submission ID trn:oid:::1:3431834634

8% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

Filtered from the Report

- Bibliography

Match Groups

- 59 Not Cited or Quoted 8%
  Matches with neither in-text citation nor quotation marks
- 0 Missing Quotations 0%
  Matches that are still very similar to source material
- 0 Missing Citation 0%
  Matches that have quotation marks, but no in-text citation
- 0 Cited and Quoted 0%
  Matches with in-text citation present, but no quotation marks

Top Sources

7%	Internet sources
4%	Publications
6%	Submitted works (Student Papers)

Integrity Flags

0 Integrity Flags for Review

No suspicious text manipulations found.

Our system's algorithms look deeply at a document for any inconsistencies that would set it apart from a normal submission. If we notice something strange, we flag it for you to review.

A Flag is not necessarily an indicator of a problem. However, we'd recommend you focus your attention there for further review.