

# Project Planning Document

---

**Version:** 1.1

**Date:** 15 November 2025

**Prepared by:** Smart Transportation Project Team

---

## Table of Contents

1. Project Overview
  2. Project Objectives
  3. Project Scope
  4. Project Deliverables
  5. Project Phases
  6. Tools & Technologies
  7. Timeline & Schedule
  8. Team Roles & Responsibilities
- 

## 1. Project Overview

Smart Transportation is a **web-based system** designed to address transportation challenges faced by **students and daily commuters**. The platform provides a **reliable, organized, and cost-effective solution** by connecting **administrators, drivers, and passengers** within a single ecosystem.

Unlike traditional systems that focus primarily on passenger convenience, Smart Transportation adopts a **driver-centric approach**. This approach **empowers drivers** with greater control and flexibility while enabling them to **earn additional income** by publishing trips, managing schedules, and serving specific routes.

The system is currently developed as a **web application**, with planned future expansion into a **cross-platform solution**, ensuring accessibility from both mobile and desktop devices.

A **basic payment gateway** has already been integrated to enable secure and simple payment processing for trip bookings. This sets the foundation for more advanced payment features in future releases.

---

## 2. Project Objectives

The primary objectives of the **Smart Transportation** system are:

1. **Reduce Waiting Times:** Enable students to access faster, more reliable transportation options and multiple booking opportunities.
  2. **Enhance Route Management:** Allow administrators to efficiently organize and monitor routes, optimizing travel schedules.
  3. **Empower Drivers:** Enable drivers to manage trips, schedules, and earnings, thereby creating an additional source of income.
  4. **Improve Passenger Experience:** Provide an intuitive platform for passengers to browse, book, and manage trips effectively.
  5. **Ensure Scalability:** Develop a web-based application with potential for cross-platform expansion.
  6. **Maintain Security and Reliability:** Guarantee secure management of user data and seamless platform operations.
  7. **Enable Payment Transactions:** Facilitate safe and efficient trip payment processing through the integrated gateway.
- 

## 3. Project Scope

**In Scope:**

1. **Trip Publishing & Booking:** Drivers can publish trips with full details—time, route, and available seats—and passengers can book seats.
2. **Route & Segment Management:** Administrators can create, edit, and manage routes, stops, and segments to optimize travel.
3. **Driver Management:** Drivers can manage profiles, vehicles, trips, and earnings.

4. **Passenger Management:** Passengers can create accounts, browse trips, book seats, and track travel history.
5. **System Notifications:** Users receive notifications for trip updates, cancellations, and confirmations.
6. **Payment Gateway (Basic Integration):** Allows passengers to make payments securely for booked trips.
7. **Future-Proof Architecture:** Designed as a web application ready for future cross-platform expansion.

#### **Out of Scope / Future Enhancements:**

1. **Advanced Payment Features:** Integration with multiple payment providers and additional payment options.
  2. **Data Analytics & Reporting:** Detailed reports on trips, bookings, and driver performance for administrators.
  3. Real-time GPS vehicle tracking.
  4. Mobile application development (planned future release).
  5. External traffic or public transportation schedule integration.
- 

## **4. Project Deliverables**

### **1. Functional Web Application**

- Trip publishing and booking system for drivers and passengers.
- Route and segment management tools for administrators.
- User account management for drivers, passengers, and administrators.
- Notification system for trip updates and cancellations.
- Basic Payment Gateway Integration for trip booking transactions.

### **2. Documentation**

- Project planning document (this document).
- Database design and ERD diagrams.

- UI/UX design wireframes and prototypes.
- Testing reports and deployment guides.

### **3. Future Deliverables (Enhancements)**

- Cross-platform mobile application.
  - Data analytics and reporting tools for administrators.
  - Integration with external services such as payments and GPS tracking.
- 

## **5. Project Phases**

Phase	Description	Deliverables
<b>1. Analysis</b>	Gather requirements, define objectives, and understand user needs.	Requirements Specification Document, Stakeholder Analysis
<b>2. Design</b>	Design system architecture, database schema, UI/UX, and payment workflow.	Database ERD, Wireframes, UI/UX Mockups, System Architecture Diagram
<b>3. Implementation</b>	Develop backend, frontend, integrate core functionalities, and implement basic payment gateway.	Functional Web Application (Core Features)
<b>4. Testing</b>	Perform unit, integration, and system testing, including payment processing verification.	Test Reports, Bug Fix Logs
<b>5. Deployment &amp; Maintenance</b>	Deploy the system to production, monitor performance, and document operational guidelines.	Deployed Application, User Guides, Maintenance Plan

---

## **6. Tools & Technologies**

- **Backend:** ASP.NET Core 9, C#
- **Frontend:** HTML5, CSS3, JavaScript (React/Blazor for future UI)
- **Database:** SQL Server
- **UI/UX Design:** Figma or Adobe XD

- **Version Control:** Git & GitHub
  - **Payment Gateway:** Basic integration (custom or third-party API)
  - **Testing Tools:** NUnit, Selenium (for future automation)
  - **Project Management:** Jira, Trello, or Excel for timeline and task tracking
- 

## 7. Timeline & Schedule

Phase	Duration	Timeline (Weeks)
Analysis	1 week	Week 1
Design	2 weeks	Week 2–3
Implementation	4 weeks	Week 4–7
Testing	1 week	Week 8
Deployment & Maintenance	1 week	Week 9

This shows the relative flow of work across 9 weeks, starting from the project kickoff (22 August).

---

## 8. Team Roles & Responsibilities

Role	Responsibilities
<b>Project Manager / Admin</b>	Oversee project progress, manage stakeholders, and ensure timelines are met.
<b>Backend Developer</b>	Develop APIs, database interactions, server-side logic, and implement payment integration.
<b>Frontend Developer</b>	Implement UI, integrate with backend, ensure responsive design, and connect payment interface.
<b>UI/UX Designer</b>	Design wireframes, mockups, user flows, and ensure intuitive navigation.
<b>Tester / QA</b>	Conduct functional, integration, and system testing; verify payment processing and report issues.

---