Bus Ticket Booking System - Project Timeline and Milestones

Project Overview

The Bus ticket booking system is a web-based platform for intercity bus ticketing in Vietnam. The project aims to achieve both MVP functionality (Criteria 1: 8.5/10.0) and advanced microservices architecture (Criteria 2: +2.5/10.0) for a total possible score of **11.0/10.0**.

Start Date: November 24, 2025

Duration: 5 weeks

End Date: December 29, 2025

> Any estimation time below is for suggestion only, it can be changed based on your team's technical skills.

WEEK 1 — Foundation and Setup

Goal: Establish core infrastructure, database design, and authentication systems for both user and admin portals.

Tasks

System & Infrastructure

- **Initialize repository with SPA framework for client side and HTTP server as server side** *(~2 hours)*
    - Set up project structure with chosen frontend framework (React/Vue/Angular) and backend framework (Node.js/Express, NestJS,

etc.)
    - **Team member:** Full-stack developer
- **Configure PostgreSQL database with initial schema** *(~1 hour)*
    - Install PostgreSQL, create database, design initial tables for users, routes, trips, bookings
    - **Team member:** Backend developer
- **Setup Redis for caching and session management (Optional)** *(~1 hour)*
    - Install and configure Redis server for session storage and caching frequently accessed data
    - **Team member:** Backend developer
- **Setup CI/CD pipeline with GitHub Actions (Optional)** *(~1 hour)*
    - Configure automated testing, building, and deployment workflows
    - **Team member:** DevOps engineer
- **Configure ESLint, Prettier, Husky for developing tools** *(~2 hours)*
    - Set up code formatting, linting rules, and pre-commit hooks for code quality
    - **Team member:** Full-stack developer
- **Create project documentation structure** *(~30 minutes)*
    - Set up README, API docs template, and development guidelines
    - **Team member:** Full-stack developer

System & Infrastructure / Auth

- **Complete user registration with email verification** *(~2 hours)*
    - Create registration form, email verification flow, and user account activation process
    - **Team member:** Backend developer
- **Implement JWT-based authentication system** *(~1 hour)*

    - Set up JWT token generation, validation, and middleware for protected routes
    - **Team member:** Backend developer
- **Implement admin authentication and authorization** *(~1 hour)*
    - Create role-based access control system to distinguish admin and regular users
    - **Team member:** Backend developer
- **Setup Google OAuth integration** *(~2 hours)*
    - Configure Google OAuth 2.0 for social login functionality
    - **Team member:** Full-stack developer
- **Create password reset functionality** *(~1 hour)*
    - Build forgot password flow with email-based reset tokens
    - **Team member:** Backend developer

Admin Portal

- **Create basic admin dashboard layout** *(~5 hours)*
    - Design and implement admin interface with navigation, overview cards, and basic CRUD operations
    - **Team member:** Frontend developer

Total estimation time spent: ~19 hours 30 minutes

WEEK 2 — Trip Management and Search

Goal: Implement core trip management features for admins and search functionality for users to enable the primary discovery flow.

Tasks

Admin Portal

- **Create trip scheduling interface for admins** *(~3 hours)*

    - Build admin interface to create, edit, and manage trip schedules with date/time selection
    - **Team member:** Frontend developer
- **Create bus assignment and scheduling logic for admins** *(~2 hours)*
    - Implement backend logic to assign buses to routes and prevent scheduling conflicts
    - **Team member:** Backend developer
- **Implement route configuration with pickup/dropoff points** *(~2 hours)*
    - Create system to define routes with multiple pickup and dropoff locations
    - **Team member:** Backend developer
- **Design and implement seat map configuration tool** *(~4 hours)*
    - Build visual tool for admins to configure bus seat layouts and pricing
    - **Team member:** Full-stack developer

User Portal / Search at Homepage

- **Create search interface with autocomplete for cities** *(~4 hours)*
    - Build user-friendly search form with city autocomplete and date picker
    - **Team member:** Frontend developer

User Portal / Search Result Page

- **Develop trip search API with filtering capabilities** *(~3 hours)*
    - Create REST API endpoints for searching trips by origin, destination, date with filters
    - **Team member:** Backend developer
- **Implement advanced filtering (time, price, bus type, amenities)** *(~4 hours)*
    - Add filter options for departure time, price range, bus types, and amenities
    - **Team member:** Frontend developer
- **Develop trip search results interface** *(~5 hours)*

  - Create results page showing trip cards with filtering, sorting, and pagination
  - **Team member:** Frontend developer
- **Add search result sorting and pagination** *(~3 hours)*
  - Implement sorting by price, time, duration and paginate results for performance
  - **Team member:** Backend developer
- **Implement trip details page** *(~4 hours)*
  - Build detailed trip view with route info, amenities, policies, and booking button
  - **Team member:** Frontend developer

Total estimation time spent: ~34 hours

WEEK 3 — Booking System and Seat Selection

Goal: Build the complete booking flow including seat selection, passenger information, and ticket generation.

Tasks

User Portal / Seat Selection

- **Create interactive seat map component** *(~6 hours)*
  - Build visual seat map with clickable seats, different seat types, and status indicators
  - **Team member:** Frontend developer
- **Implement seat locking mechanism** *(~4 hours)*
  - Create temporary seat reservation system to prevent double bookings during checkout
  - **Team member:** Backend developer

- **Develop seat availability updates** *(~3 hours)*
    - Implement WebSocket or polling to show real-time seat status changes
    - **Team member:** Full-stack developer
- **Create seat selection validation logic** *(~2 hours)*
    - Add validation to prevent selecting unavailable seats and enforce seat limits
    - **Team member:** Backend developer

User Portal / Booking Flow

- **Build passenger information collection forms** *(~3 hours)*
    - Create forms to collect passenger details (name, ID, phone) for each selected seat
    - **Team member:** Frontend developer
- **Implement booking creation and management** *(~6 hours)*
    - Build backend API to create bookings, manage booking states, and handle expiration
    - **Team member:** Backend developer
- **Create booking summary and review interface** *(~3 hours)*
    - Design booking review page showing trip details, passengers, and total cost
    - **Team member:** Frontend developer
- **Develop booking history and management dashboard** *(~4 hours)*
    - Build user dashboard to view, modify, and cancel existing bookings
    - **Team member:** Full-stack developer

User Portal / Guest Services

- **Implement guest checkout flow without registration** *(~3 hours)*

    - Allow users to book tickets without creating an account, collecting minimal required info
    - **Team member:** Full-stack developer
- **Create guest booking lookup system** *(~2 hours)*
    - Build system for guests to retrieve bookings using reference phone number or email
    - **Team member:** Backend developer
- **Setup booking reference generation** *(~1 hour)*
    - Create unique, user-friendly booking reference number generation system
    - **Team member:** Backend developer

User Portal / Ticketing

- **Create e-ticket download/sharing or email delivery (choose 1 to implement)** *(~2 hours)*
    - Implement download functionality/sharing ticket functionality
    - Implement automatic email delivery of e-tickets
    - **Team member:** Backend developer
- **Design e-ticket template with branding** *(~1 hour)*
    - Design professional e-ticket layout with company branding and essential information
    - **Team member:** Frontend developer

Total estimation time spent: ~40 hours

WEEK 4 — Payment Integration and Notifications

Goal: Integrate payments, notifications, and post-booking management to complete the transaction lifecycle.

Tasks

User Portal / Payments

- **Integrate PayOS payment gateway** *(~3 hours)*
  - Set up PayOS API integration for processing credit card and digital wallet payments
  - **Team member:** Backend developer
- **Implement payment webhook handling** *(~3 hours)*
  - Create webhook endpoints to receive payment status updates from payment gateways
  - **Team member:** Backend developer
- **Create payment confirmation and failure flows** *(~2 hours)*
  - Build user interfaces for successful payments and error handling for failed payments
  - **Team member:** Frontend developer

User Portal / Notifications

- **Setup email service** *(~1 hour)*
  - Configure email service provider (SendGrid/AWS SES) for sending transactional emails
  - **Team member:** Backend developer
- **Create email templates for booking confirmations** *(~2 hours)*
  - Design and implement HTML email templates for booking confirmations and receipts
  - **Team member:** Frontend developer
- **Setup trip reminder notifications** *(~2 hours)*
  - Create scheduled job system to send reminder emails/SMS before trip departure
  - **Team member:** Backend developer
- **Create notification preferences management** *(~2 hours)*

    - Build user interface to manage email and SMS notification preferences
    - **Team member:** Frontend developer

User Portal / Management

- **Create booking modification functionality** *(~4 hours)*
    - Allow users to modify passenger details and change seats (if available)
    - **Team member:** Full-stack developer
- **Setup automated booking expiration** *(~1 hour)*
    - Implement background job to automatically cancel unpaid bookings after timeout
    - **Team member:** Backend developer

Admin Portal

- **Create revenue analytics dashboard** *(~3 hours)*
    - Build admin dashboard showing revenue metrics, charts, and financial reports
    - **Team member:** Frontend developer
- **Implement booking analytics and reporting** *(~2 hours)*
    - Create analytics for booking trends, popular routes, and conversion rates
    - **Team member:** Backend developer

Total estimation time spent: ~32 hours

WEEK 5 — Final Features and Project Wrap-up

Goal: Implement advanced features, optimize performance, and deploy the system to production.

Tasks

User Portal / AI Assistant

- **Setup OpenAI API integration** *(~1 hour)*
  - Configure OpenAI API credentials and create service layer for AI interactions
  - **Team member:** Backend developer
- **Create chatbot interface component** *(~2 hours)*
  - Build chat widget with message history, typing indicators, and responsive design
  - **Team member:** Frontend developer
- **Implement natural language trip search** *(~3 hours)*
  - Create AI prompts to understand user queries and convert to search parameters
  - **Team member:** Full-stack developer
- **Enable booking through chatbot** *(~3 hours)*
  - Allow users to complete entire booking process through conversational interface
  - **Team member:** Full-stack developer
- **Create FAQ handling system** *(~2 hours)*
  - Train chatbot to answer common questions about policies, routes, and booking process
  - **Team member:** Backend developer

User Portal / Feedback

- **Create user review and rating interface** *(~2 hours)*
  - Build frontend components for rating trips and displaying reviews to other users
  - **Team member:** Frontend developer
- **Implement feedback and rating system** *(~2 hours)*

  - Create backend system for users to rate trips and leave feedback after completion
  - **Team member:** Backend developer

System & Infrastructure / Advanced

- **Implement API gateway with Kong/Nginx** *(~2 hours)*
  - Set up API gateway for routing, rate limiting, and load balancing across microservices
  - **Team member:** DevOps engineer
- **Setup service discovery with Consul/Kubernetes** *(~2 hours)*
  - Configure service registry and discovery for dynamic service communication
  - **Team member:** DevOps engineer
- **Create concurrent booking handling system** *(~2 hours)*
  - Implement distributed locking and conflict resolution for simultaneous booking attempts
  - **Team member:** Backend developer
- **Setup multiple authentication methods** *(~1 hour)*
  - Configure OAuth providers (Google, Facebook) and phone number authentication
  - **Team member:** Backend developer

System & Infrastructure / QA

- **Write comprehensive unit tests** *(~3 hours)*
  - Create unit tests for critical business logic with minimum 80% code coverage
  - **Team member:** Full-stack developer
- **Implement integration testing** *(~2 hours)*
  - Test API endpoints and database interactions with automated test suite

    - **Team member:** Backend developer
- **Conduct end-to-end testing** *(~2 hours)*
    - Test complete user workflows from search to booking completion using automation tools
    - **Team member:** Frontend developer
- **Performance testing and optimization** *(~2 hours)*
    - Load test the application and optimize bottlenecks for concurrent users
    - **Team member:** DevOps engineer
- **Security testing and vulnerability assessment** *(~1 hour)*
    - Run security scans and test for common vulnerabilities (SQL injection, XSS, etc.)
    - **Team member:** Backend developer

System & Infrastructure / Deployment

- **Setup production environment on cloud** *(~2 hours)*
    - Configure cloud infrastructure (AWS/GCP/Azure) with proper security and scaling settings
    - **Team member:** DevOps engineer
- **Configure monitoring or logging** *(~1 hour)*
    - Set up application monitoring, error tracking, and centralized logging system
    - **Team member:** DevOps engineer
- **Final deployment and go-live** *(~1 hour)*
    - Deploy application to production, perform final checks, and announce go-live
    - **Team member:** Full-stack developer

Total estimation time spent: ~37 hours

Technical Requirements Checklist

Criteria 1 (8.5/10.0) - MVP Features

- □ Fully functional bus ticket booking web application
- □ Integrated payment functionality (MoMo, Zalopay)
- □ Guest checkout support
- □ AI chatbot integration for booking assistance
- □ Public deployment

Criteria 2 (+2.5/10.0) - Advanced Features

- □ Microservices architecture implementation
- □ CI/CD pipeline with automated testing
- □ Concurrent booking handling with seat locking
- □ Saga pattern for distributed transaction management
- □ Multiple authentication methods (email, OAuth, phone)

Risk Mitigation Strategies

Technical Risks

Payment gateway integration delays

- Start integration early in Week 4, have fallback options

Microservices complexity

- Begin with monolithic approach, refactor to microservices

Performance issues

- Implement caching and database optimization from Week 2

Timeline Risks

Feature scope creep

- Prioritize MVP features first, advanced features second

Testing time

- Allocate dedicated testing time in Week 5

Deployment issues

- Setup staging environment early for testing

Team Coordination

- **Daily standups:** 15-minute daily sync meetings
- **Weekly reviews:** End-of-week progress review and planning
- **Code reviews:** Mandatory peer review for all code changes

Success Metrics

Week 1 Success Criteria

- All development environments setup and functional
- Authentication system working with user registration/login
- Basic admin interface operational
- CI/CD pipeline configured

Week 2 Success Criteria

- Trip search functionality working with filters
- Admin can create and manage routes/buses
- Database performance optimized for search queries

Week 3 Success Criteria

- Complete booking flow functional
- Seat selection with real-time updates working
- E-ticket generation/ticket delivery operational

- Guest checkout flow completed

Week 4 Success Criteria

- All payment gateways integrated and tested
- Notification system sending emails and SMS
- Booking management (cancel/modify) working
- Admin analytics dashboard functional

Week 5 Success Criteria

- AI chatbot operational for search and booking
- Microservices architecture implemented
- All testing completed with >70% test coverage
- Production deployment successful

Communication Plan (Suggestion)

Internal Communication

- **Daily standups:** 9:00 AM (15 minutes)
- **Weekly planning:** Monday 2:00 PM (1 hour)
- **Sprint reviews:** Friday 4:00 PM (1 hour)
- **Technical discussions:** As needed via Slack/Discord/Google Meet

Documentation

- API documentation with Swagger/OpenAPI
- User guides for admin and end-users
- Technical architecture documentation
- Deployment and maintenance guides

Untitled