## **DECLARATION OF THE STUDENT**

We hereby declare that the project entitled is an outcome of our own efforts under the guidance of **Dr. Annu Mishra**. The project is submitted to the Sharda University for the partial fulfilment of the Bachelor of Technology Examination 2024-25.

We also declare that this project report has not been previously submitted to any other university.

**Sumit Gupta**

**(2022498440)**

**CERTIFICATE**

This is to inform that **Sumit Gupta** of Sharda University has successfully completed the project work titled Frontend Bus Ticket Booking at Aasakya Digital Technologies Pvt Ltd in partial fulfilment of the Bachelor of Technology Examination 2024-2025 by Sharda University.

This project report is the record of authentic work carried out by them during the period from

**Sumit Gupta**

**(2201010721)**

**Dr. Annu Mishra**

**(**Asst. Professor)

**Dr. Sudeep Varshney**

Professor & Head, Department of CSE

**LIST OF TABLES**

|  |  |  |
| --- | --- | --- |
| **Table No.** | **Title** | **Page No.** |
| 1.1 | Hardware Requirements for Bus Ticket Booking | 10 |
| 1.2 | Software Requirements for Bus Ticket Booking | 11 |
| 2.1 | Comparison of Existing Ticketing | 16 |
| 3.1 | |  | | --- | |  |   Component-Based | 19 |
| 3.2 | Tools And Technologies Used | 20 |
| 3.3 | Modules and Responsibilities | 21 |
| 4.1 | Feature Implementation Overview | 23 |

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **Figure No.** | **Title** | **Page No.** |
| 1.1 | Real-World Testing Workflow Observed During Internship | 12 |
| 3.1 | Wireframe :Home/Landing | 21 |
| 3.2 | UI Screenshot: Booking Form | 22 |
| 4.1 | Responsive Layout | 24 |
| 4.2 | Seat Selection Panel | 24 |
| 4.3 | Booking Summary/Confirmation Card | 25 |
| 4.4 | Accessibility Checklist Snapshot | 25 |

**ABSTRACT**

In the public transportation domain, a clear and responsive interface for bus ticket booking can significantly improve user experience and reduce friction during trip planning. **Bus Ticket Booking UI** is a **front-end–only** project developed using **HTML5, CSS3, and Bootstrap 5**. The interface enables users to enter journey details (source, destination, date), preview available buses, select seats (prototype UI), and view a booking summary/confirmation screen.

Designed with a **mobile-first** approach and consistent visual hierarchy, the UI offers a clean layout, component reuse, and beginner-friendly code. This report documents the **problem definition**, **specifications**, **literature survey**, **design & implementation**, **results**, and **future scope**. While the system is intentionally **static** (no backend/database), it is architecture-ready for **API integration**, **authentication**, and **payment gateway** additions.

**ACKNOWLEDGEMENT**

I would like to express my deepest appreciation to all those who provided me the possibility to complete this report. Apart from the efforts of myself, the success of any project depends largely on the encouragement and guidelines of many others. We take this opportunity to express my gratitude to the people who have been instrumental in the successful completion of this project. We would like to show my greatest appreciation to **Dr. Annu Mishra**. We can’t say thank you enough for her/his tremendous support and help. We feel motivated and encouraged every time we attend her meeting. Without her encouragement and guidance this project would not have materialized. The guidance and support received from all the members who contributed and who are contributing to this project, was vital for the success of the project. We are grateful for their constant support and help. Besides, we would like to thank the authority of Sharda University for providing us with a good environment and facilities to complete this project. Finally, an honourable mention goes to our families and friends for their understandings and supports on us in completing this project. Without helps of the particular that mentioned above, we would face many difficulties while doing this.

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **Sr. No.** | **Contents** | **Page No.** |
|  | Declaration of the Student | i |
|  | Certificate of the Guide | ii |
|  | List of Tables | iii |
|  | List of Figures | iv |
|  | Abstract | v |
|  | Acknowledgement | vi |
| 1 | **INTRODUCTION**   1. Problem Definition 2. Hardware Specification 3. Software Specification 4. Motivation 5. Objectives 6. Contributions 7. Summary | 9-14 |
| 2 | **LITERATURE SURVEY**   1. Related Work   Summary | 14-16 |
| 3 | **DESIGN AND IMPLEMENTATION**   1. Methodology 2. Design 3. Implementation   Summary | 17-22 |
| 4 | **RESULT AND DISCUSSIONS**   1. Results 2. Discussion   Summary | 23-26 |
| 5 | **CONCLUSION**   1. Conclusion 2. Limitations 3. Future Scope   Summary | 27-28 |
| 6 | **REFERENCES** | 29 |

**INTRODUCTION**

**1.1 Problem Definition**

With rapid urbanization and frequent intercity travel, passengers increasingly depend on online booking platforms to plan journeys. Despite this trend, many existing bus ticketing interfaces remain cluttered, non-intuitive, or poorly optimized for mobile devices. Common problems include confusing seat-selection flows, excessive information density, unclear call-to-action elements, and inconsistent visual hierarchy. These issues increase user effort, raise abandonment rates during booking, and degrade overall user satisfaction. The Bus Ticket Booking UI project aims to address these shortcomings by delivering a clean, mobile-first front-end prototype that prioritizes clarity, accessibility, and a task-focused booking flow. The prototype demonstrates how a minimal, well-structured UI can reduce cognitive load and make the booking process faster and more reliable for end users.

**1.2 Hardware Specification**

The following table lists recommended hardware for development and testing of the front-end prototype:

Table 1.1 Hardware Specification

|  |  |
| --- | --- |
| **Component** | **Specification** |
| Processor | Intel Core i5 / AMD Ryzen 5 or higher |
| RAM | Minimum 8 GB (16 GB recommended) |
| Storage | 256 GB SSD or larger |
| Display | 13" or larger, Full HD (1920×1080) |
| Network | Broadband internet for remote testing and API work |

Notes: The UI is front-end focused and can be developed and previewed offline; however, network access is required for testing integrations or CDN-hosted libraries.

**1.3 Software Specification**

The project was implemented and tested using the software stack shown below:

Table 1.2 Software Specification

|  |  |
| --- | --- |
| **Software / Tool** | **Version / Purpose** |
| Operating System | Windows 10 / Ubuntu 20.04 |
| Code Editor | Visual Studio Code |
| Browsers | Chrome, Firefox, Edge (for testing) |
| Front-end Tech | HTML5, CSS3, Bootstrap 5 |
| Version Control | Git (optional) |
| Assets | Google Fonts, Bootstrap Icons (optional) |

**1.4 Motivation**

The motivation behind this project stems from both industry observations and learning objectives. At internships and in the field, a recurring issue is that users abandon booking processes when interfaces are confusing or not mobile-optimized. As a student and front-end intern, the goal was to design a prototype that demonstrates practical UI/UX improvements—streamlined forms, prominent search actions, card-based bus listings, and a simple seat-selection prototype—while practicing modern front-end techniques and responsive design principles. This project also serves as a foundation for integrating real backends and payment flows in future work.

**1.5 Objectives**

The primary objectives for the Bus Ticket Booking UI are:

* Design a **mobile-first**, responsive booking interface using HTML5, CSS3, and Bootstrap 5.
* Implement a clear **booking flow**: journey input → bus listings → seat selection (prototype) → booking summary.
* Ensure accessibility basics: proper labels, keyboard focus, and readable contrast.
* Create modular HTML/CSS structure to allow easy integration with APIs and backend services later.
* Produce documentation and weekly progress records suitable for internship evaluation.

**1.6 Contributions**

This internship project contributes the following deliverables:

* A complete static front-end prototype for a bus ticket booking flow with modular components (Navbar, Hero/Search, Booking Form, Bus Cards, Seat Panel mock, Summary, Footer).
* Responsive design patterns and reusable Bootstrap-based components.
* Clear, well-documented code organization and a README describing integration points for future API and backend work.
* A week-by-week diary and testing notes covering responsiveness checks and cross-browser validation.

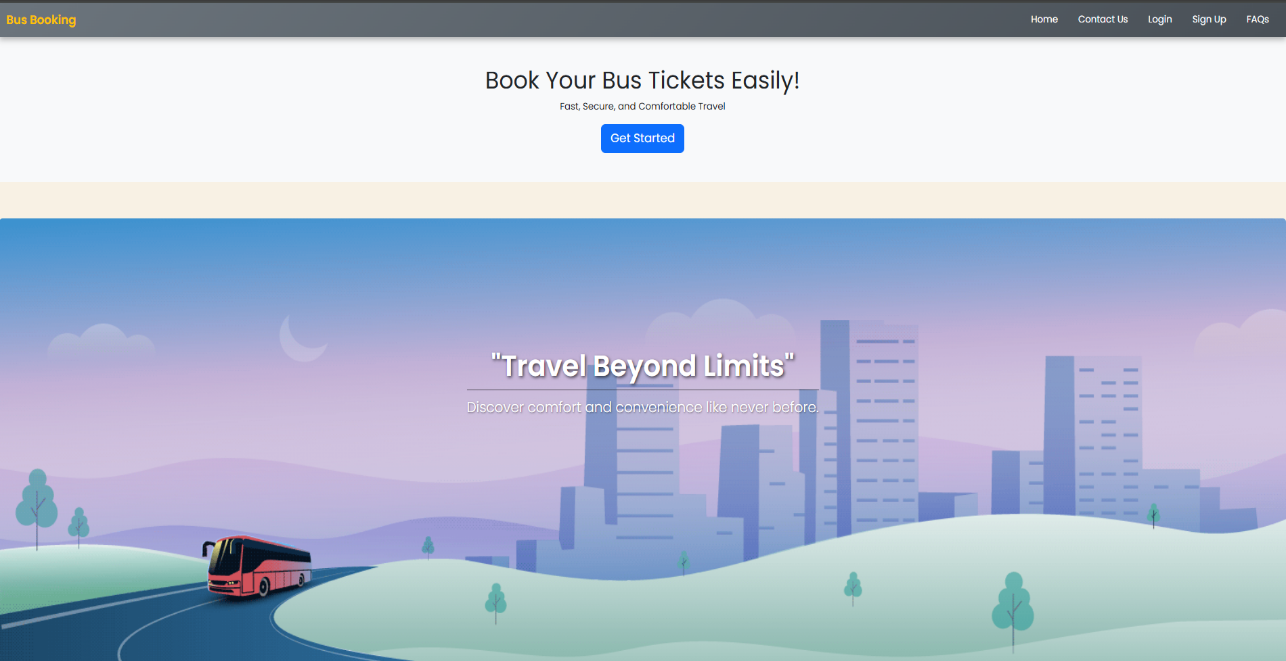


Fig1.1 Home Page

**LITERATURE SURVEY**

**2.1 Related Work**

Online bus ticketing and travel platforms have matured significantly over the past decade. Established commercial portals such as redBus, AbhiBus, and travel aggregators (e.g., MakeMyTrip) offer end-to-end booking experiences that combine search, filters, interactive seat maps, payment processing, and booking history. These systems are feature-rich and optimized for scale, but they often include complex flows, extensive server-side logic, and monetization-driven UI elements (promotions, upsells) that can clutter the booking path.

From a front-end design and prototyping perspective, many open-source UI templates and design systems (Bootstrap examples, Tailwind UI kits, and community dashboard templates) highlight best practices for responsive forms, card-based listings, and accessibility considerations. These resources emphasize modular components, consistent spacing, and mobile-first layouts—principles adopted in this project to produce a clean and maintainable front-end prototype.

Several academic and developer-oriented projects focus on isolated aspects of ticketing systems: seat-selection algorithms, route-optimization, and accessibility audits for booking flows. While such projects provide deep insights into individual subsystems, a lightweight front-end prototype that emphasizes usability and clarity (without backend complexity) is valuable for rapid testing and iterative UX improvements.

**Table 2.1 — Comparison of Typical Ticketing UIs**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **System / Template** | **Complexity** | **Seat UI** | **Mobile Optimization** | **Primary Use** |
| **redBus / AbhiBus** | **High** | **Interactive grid** | **High** | **Production booking (full-stack)** |
| **MakeMyTrip (bus module)** | **High** | **Interactive + filters** | **High** | **Aggregator with payments** |
| **Open-source Bootstrap templates** | **Low–Medium** | **Static or sample** | **Medium** | **Rapid prototyping / learning** |
| **Academic seat-selection projects** | **Medium** | **Algorithmic focus** | **Low** | **Research / algorithm testing** |
| **This Project (Bus Ticket Booking UI)** | **Low** | **Static mock (prototype)** | **High (mobile-first)** | **Front-end prototype for UX validation** |

**DESIGN AND IMPLEMENTATION**

**3.1 Methodology**

The development followed a phased, iterative front-end methodology focused on rapid prototyping and continuous refinement:

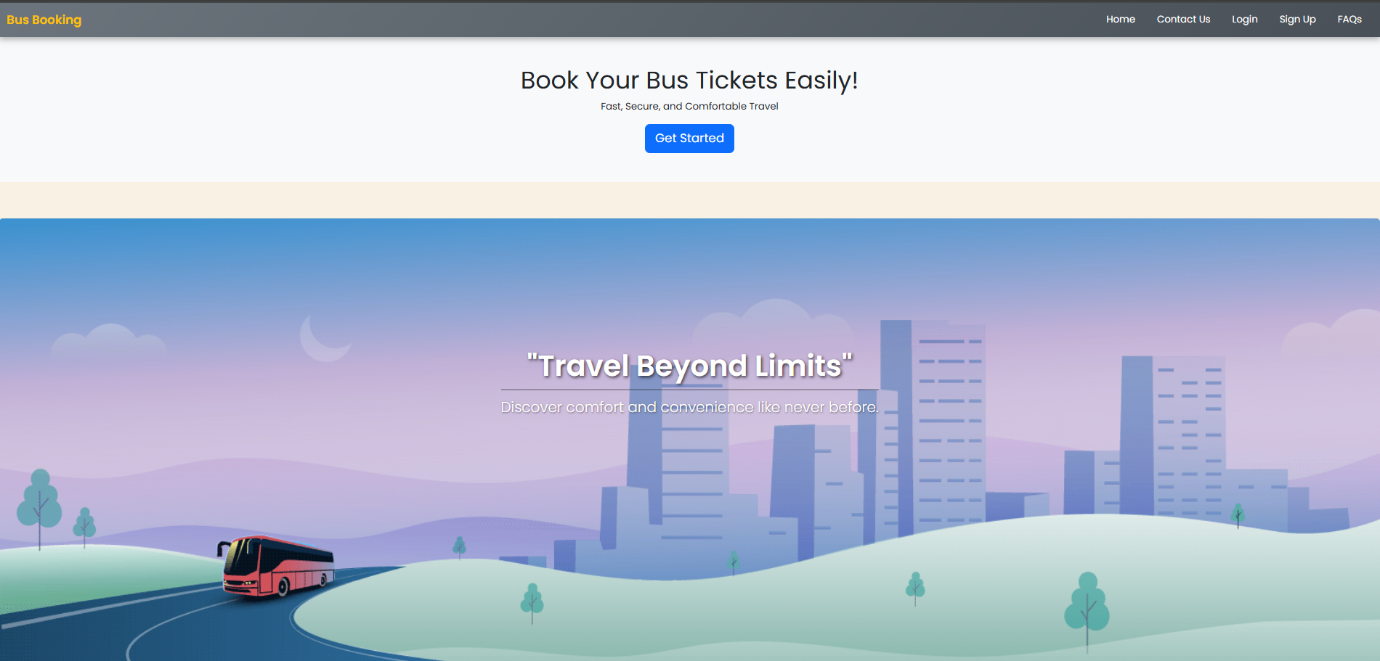
1. Requirement Understanding & Wireframing: Collected basic functional requirements (search, listings, seat prototype, summary) and produced low-fidelity wireframes to map the booking flow.
2. Static HTML Structure: Implemented semantic HTML sections (header, main, section, footer) to ensure accessible document structure and straightforward component mapping.
3. Bootstrap-based Layouting: Used Bootstrap 5 grid and utility classes to quickly create responsive layouts and consistent spacing.
4. Styling & Polish: Added custom CSS for brand colors, typography tuning, hover/focus states, and minor animations for feedback (button ripples, card hover).
5. Responsiveness & Testing: Tested the UI at common breakpoints (360px, 768px, 1024px) and iteratively adjusted spacing, font sizes, and control sizes for touch targets.
6. Peer/Mentor Review: Incorporated feedback on contrast, label clarity, and form grouping to improve usability.

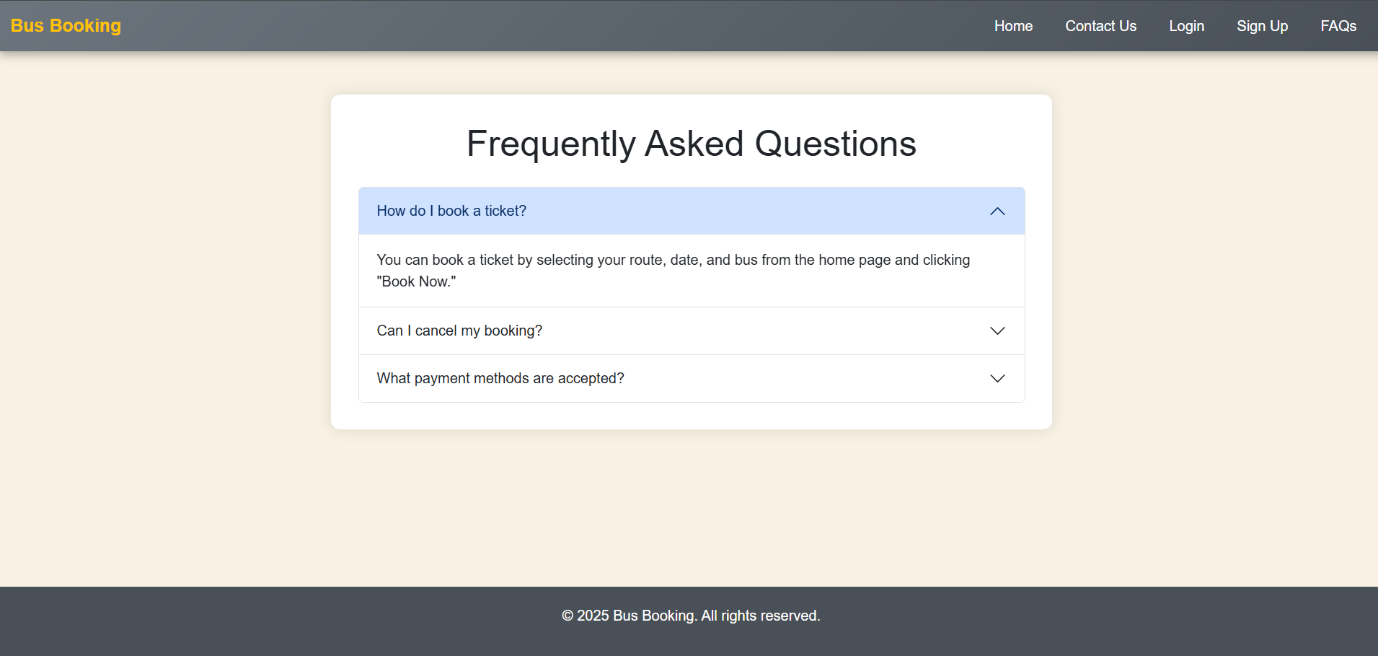
This approach prioritized a mobile-first stance—building for the smallest screen and scaling up—ensuring the booking flow remained clear on constrained viewports.

**3.2 Design**

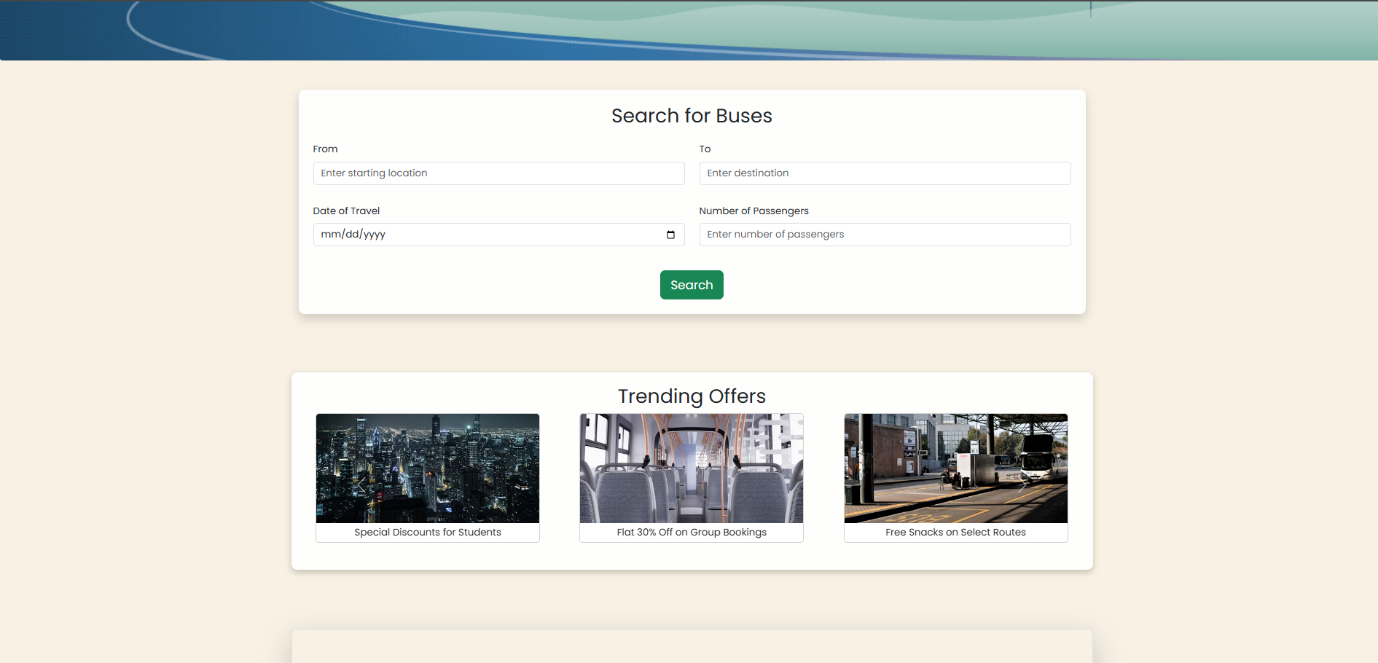
The UI design emphasizes clarity, minimalism, and stepwise progression through the booking task. Main visual and interaction elements:

* Navbar: Compact brand area with a simple account/help link.
* Hero / Quick Search: Prominent search area allowing source, destination, date, and passenger inputs; primary CTA (Search Buses) is visually dominant.
* Bus Listings: Card-based layout showing operator, departure/arrival times, duration, fare, and a primary “Select Seats” button. Cards are designed for quick scanning with key information left-aligned and fare on the right.
* Seat Panel (Prototype): Static grid that visually represents seat rows and availability states (available / blocked / selected). Visual legend included for clarity.
* Booking Summary: Compact card that lists chosen bus, selected seats, passenger count, fare breakdown, and a final Confirm CTA.
* Footer: Contact info and links.

Figure placeholders:  
  
[Fig 3.1 — Wireframe: Home / Search]



[Fig 3.2 —page]

  
[Fig 3.3 — Panel Prototype]

**3.3 Implementation**

HTML & Structure

* Semantic HTML elements used for accessibility and logical DOM flow.
* Form controls include label tags, aria attributes where applicable, and clear placeholder text to aid usability.

CSS & Theming

* Bootstrap 5 forms, grid, cards, and buttons form the baseline.
* A small custom stylesheet adjusts brand colors, spacing utilities, and responsive text scaling (rem units).

JavaScript (UI Hooks)

* Minimal JS was included for UI enhancement (toggling seat selection visuals, opening modal confirmation). All interactive behavior is client-side and non-persistent (no backend).
* Placeholders and data-\* attributes are added to key elements for easy future wiring to APIs.

Table 3.1 — Component-Based Architecture

|  |  |
| --- | --- |
| Component | Responsibility |
| Navbar | Branding, navigation, quick links |
| Hero/Search | Journey input, date, passenger selector |
| BusCard | Bus info, fare, select seats CTA |
| SeatPanel | Seat map mockup and selection visuals |
| SummaryCard | Fare breakdown, selected seats, confirm button |
| Footer | Contact, policies |

Table 3.2 — Tools & Libraries

|  |  |
| --- | --- |
| Tool / Library | Purpose |
| HTML5 / CSS3 | Structure & styling |
| Bootstrap 5 | Responsive grid & components |
| VS Code | Development environment |
| Chrome DevTools | Debugging & responsive testing |
| Git | Versioning (optional) |

**RESULT AND DISCUSSIONS**

**4.1 Results**

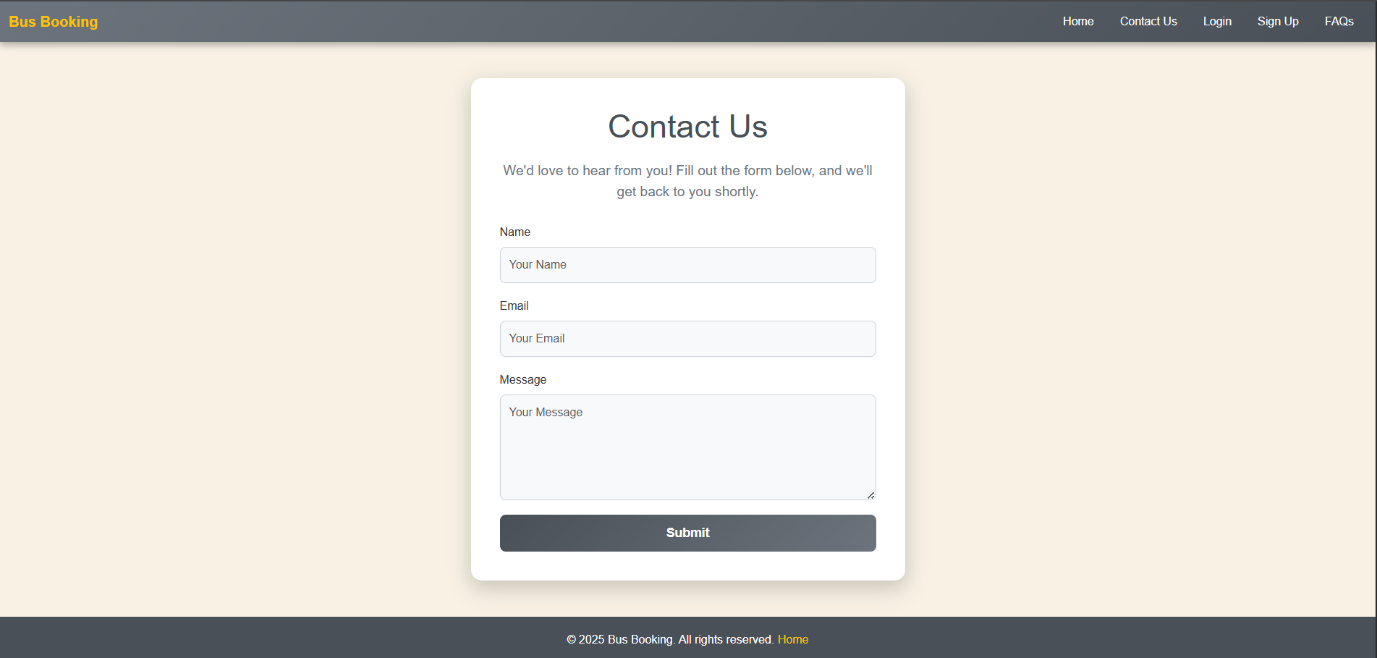
The Bus Ticket Booking UI prototype successfully implements the planned front-end features and meets the project’s usability and responsiveness goals. Key delivered components include:

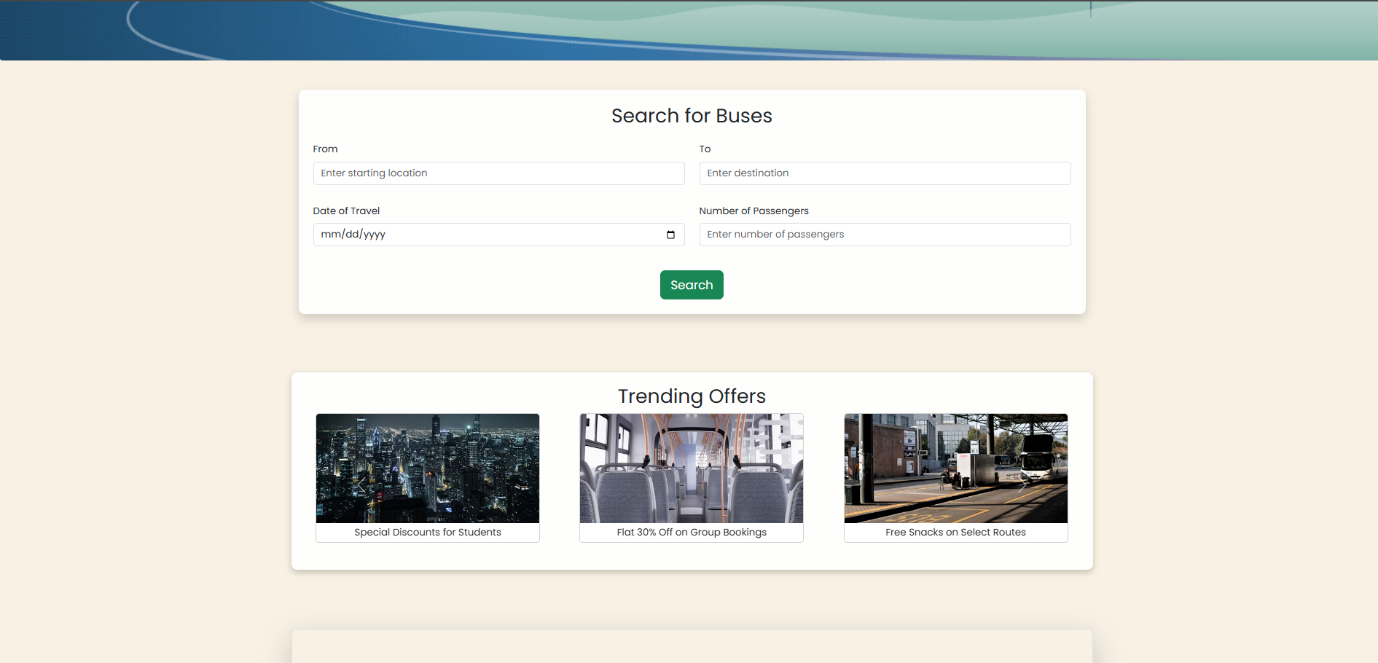
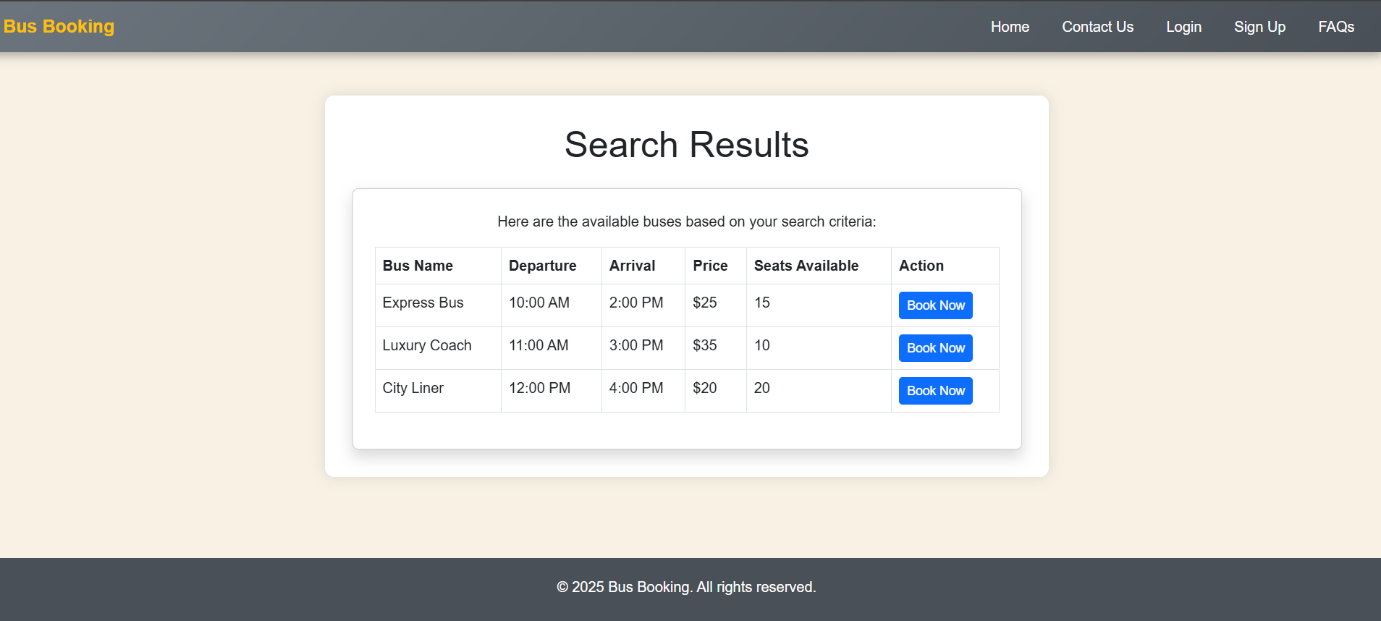
* Booking Form: Source, destination, date, and passenger inputs with clear labels and validation hints (UI-only).
* Bus Listings: Card-based presentation showing operator, timings, duration, and fare with a primary CTA to select seats.
* Seat Panel (Prototype): Static seat grid illustrating available, blocked, and selected states with a legend.
* Booking Summary: Compact fare breakdown, selected seats, and confirm CTA for final review.
* Responsiveness: Layout adapts effectively between mobile (≤ 480px), tablet (~768px), and desktop (≥1024px) breakpoints.

Table 4.1 — UI Component Testing Summary

|  |  |  |
| --- | --- | --- |
| Component | Test Performed | Result / Notes |
| Booking Form | Input focus, label association | Passed — labels and placeholders clear |
| Bus Listings | Card stacking and spacing | Passed — cards reflow correctly |
| Seat Panel | Visual states and selection demo | Passed — static mock states visible |
| Summary Card | Content truncation and wrapping | Passed — information readable |
| Responsiveness | 360px / 768px / 1024px breakpoints | Passed — layout consistent |
| Cross-Browser | Chrome, Firefox, Edge | Passed — minor style tweaks required |

Figure placeholders:

  
[Fig 4.1 — Responsive Layout (mobile/tablet/desktop)]

  
[Fig 4.2 —Selection Panel (prototype)]

[Fig 4.3 — Booking Summary Card]

**4.2 Discussion**

The prototype confirms that a well-structured Bootstrap-based front-end can deliver a clear booking flow with minimal development overhead. Bootstrap utilities provided consistent spacing and a reliable grid system; custom CSS handled branding and micro-interactions. Major observations from testing and review sessions:

* Mobile-first Advantages: Designing from smallest to largest screens forced prioritization of core actions (search and CTA), reducing clutter and improving conversion-focused layout.
* Clarity over Features: Focusing on essential tasks (search → select → confirm) produced a simpler user journey with fewer cognitive demands than feature-heavy production sites.
* Accessibility Considerations: Adding explicit label elements, sufficient color contrast, and larger touch targets increased perceived usability during manual tests. However, automated accessibility audits (e.g., AXE) should be performed in future iterations to systematically catch issues.
* Integration Readiness: All form inputs and interactive elements include semantic names/data-\* attributes to ease mapping to backend APIs. This design choice reduces future integration effort.

Challenges & Limitations Observed:

* Seat selection is a static mock; the lack of dynamic state management (no persistent data or API) prevents real booking flows.
* Browser rendering differences required minor CSS overrides for consistent spacing, particularly on legacy Edge versions.
* No authentication or payment flows were included in scope, limiting end-to-end validation.

**4.3 Recommendations & Next Steps**

To evolve the prototype into a production-ready module, recommended priorities are:

1. API Integration: Connect to a backend for real-time bus data, seat inventory, and booking operations.
2. Dynamic Seat Management: Implement JavaScript-driven seat states with server-side locking to prevent race conditions.
3. Form Validation & Error Handling: Add client- and server-side validation, plus graceful error/timeout states.
4. Accessibility Audit: Run automated checks and remediate ARIA roles, keyboard navigation, and screen reader flow.
5. Performance Optimization: Minify assets, use optimized images, and leverage caching/CDN for faster load times.

**CONCLUSION**

**5.1 Conclusion**

The Bus Ticket Booking UI internship project successfully produced a clean, mobile-first front-end prototype for bus ticket reservations using HTML5, CSS3, and Bootstrap 5. The implemented modules (Navbar, Hero/Search, Booking Form, Bus Listing Cards, Seat Panel mock, Booking Summary, Footer) demonstrate core UI/UX principles: clarity, visual hierarchy, readable forms, and responsive behavior across typical device breakpoints. The exercise strengthened practical front-end skills—semantic markup, Bootstrap layout design, CSS theming, and basic client-side interactivity—while producing a maintainable codebase ready for further back-end wiring.

**5.2 Limitations**

* No Backend Integration: The UI is static; there is no server-side API for routes, seat inventory, or bookings.
* Static Seat Selection: Seat panel is a visual prototype and does not support persistence, locking, or concurrent booking checks.
* No Payment or Auth: Authentication, user profiles, and payment flows are out of scope for this prototype.
* Testing Scope: Testing focused on manual responsiveness and cross-browser appearance; automated tests and performance profiling were not implemented.

**5.3 Future Scope**

The project is intentionally structured to make next steps straightforward. Recommended extensions:

1. API Integration: Implement REST endpoints (or GraphQL) for route search, seat availability, fare calculation, and booking confirmation.
2. Dynamic Seat Management: Build client-server seat locking, real-time updates (WebSockets) to prevent double bookings.
3. Authentication & Profiles: Add user accounts, booking history, and saved passenger details to streamline repeat bookings.
4. Payment Gateway: Integrate secure payment processors (Razorpay, Stripe, etc.) and handle transaction states.
5. Accessibility & i18n: Conduct automated accessibility audits (axe/core), add ARIA attributes, improve keyboard navigation, and add language support.
6. Performance: Optimize assets, lazy-load images, and add caching strategies when integrating APIs.

**REFERENCES**

1. Bootstrap Documentation — https://getbootstrap.com
2. Mozilla Developer Network (MDN) — <https://developer.mozilla.org> (HTML, CSS, JavaScript references)
3. W3Schools — <https://www.w3schools.com> (front-end tutorials and examples)
4. Google Material Design Guidelines — https://m3.material.io (layout and spacing principles)
5. freeCodeCamp — https://www.freecodecamp.org (responsive design articles)
6. Stack Overflow — <https://stackoverflow.com> (troubleshooting & examples)
7. “Designing Interfaces” (Scott Berkun) — for UX layout patterns and CTA placement (recommended reading)