OPTIMIZING CAMPUSWAY: REAL-TIME BUS BOOKING AND DATA-DRIVEN ALLOCATION

Ali Md Ibrahim[1]
[1]Department of Computer Science and Engineering, Leading University, Sylhet 3112, Bangladesh

Suvasish Banerjee[2]
[2]Department of Computer Science and Engineering, Leading University, Sylhet 3112, Bangladesh

Zaima Muazzama[3]
[3]Department of Computer Science and Engineering, Leading University, Sylhet 3112, Bangladesh

Md. Jehadul Islam Mony[4*]
[*4]Department of Computer Science and Engineering, Leading University, Sylhet 3112, Bangladesh

*Corresponding

Abstract
Campus transport plays a vital role in ensuring timely and reliable mobility for students, faculty and staff within the university environment. CampusWay collects transportation data, which can be analyzed to identify gaps in the system, and if integrated with machine learning, can provide predictive insights for improving campus transport efficiency. An Incremental Software Development Model was used to develop the web application. In this model, the system was built in successive increments, with each increment adding functional modules to the application, while initial prototypes designed in Figma helped visualize the user interface and gather feedback before implementation.According to the design, the CampusWay system comprises of three main modules, namely, the Login & Registration Module, the User Booking Module and the Administrator's Module. In addition, Use CaseDiagrams, Class Diagrams and Data Flow Diagrams were presented to illustrate data interactions within the system. The web application was implemented using the Laravel framework with PHP as the backend language, MySQL as the database-management system, and Blade templates with HTMLand CSS for the frontend design. The application was hosted on a local server environment for development and testing purposes. Also the screenshots of selected webpages were used to demonstrate the effectiveness of the developed web application in accomplishing the various user requirements for the smart bus transport system.

Keywords: *CampusWay, University Bus Booking, Web Application, Incremental Development, Transport Management System.*

Published: August 23, 2025

I.Introduction

The efficient management of university transportation is significant to ensuring students get to their classes, exams, and extracurricular activities on time. Many universities still work with outdated methods like handwritten records or simple digital forms [1]. These systems often cause ongoing issues such as overbooked seats, poorly allocated buses, and limited transparency between students and administrators. As a result, both students and staff face challenges in smooth and reliable transportation. This problem directly affects student satisfaction and their daily performance at the university. Students suffer from unavailable seats, no record of past bookings, and no way to cancel tickets. On the other hand, Administrators face the struggle to manage schedules, allocate buses effectively, and generate useful reports [2]. Without a proper system, resources are often mismanaged, overcrowded routes get insufficient buses due to the need for students, while less busy routes have more than enough. Previous digital systems have mostly offered basic booking platforms or static timetables [3]. While these reduce paperwork, they don't provide real-time updates, analytics, or smart features to optimize bus allocation. Most of the systems fail to give useful summaries, like weekly, monthly, semester-wise, or yearly reports, which are vital for planning [4].

To solve these problems, this study introduces CampusWay, a web-based bus booking and management system that solves the seat issue for students, provides a proper bus allocation suggestion system for administrators, and gives transparency between students and administrators. Students can log in and register, manage their password, perform real-time seat booking, track their booking history, perform ticket cancellations, update contact information, and secure logout. Administrators get a centralized dashboard with four main features: (1) managing bus schedules, (2) tracking all student bookings, (3) reviewing daily route schedules and daily booked seats (4) monitoring summarized data for weeks, months, semesters, and years. The system also includes pages for complete booking lists, bus management, add driver, add bus, and daily scheduling.

One of the best features of CampusWay is its calculation module with a suggestion engine for future planning. Using historical booking data, it can recommend how many buses are needed on each route at different times. This ensures busy routes get enough buses while avoiding waste on less busy routes.

In short, this study contributes by:

- Development of a detailed web-based platform for real-time university bus booking for students and data data-driven bus allocation system for administrators.
- A dual interface for students and administrators for a smooth experience, transparency, and structured operations.
- Implementation of reporting and analytics features, including weekly, monthly, semester-wise, and yearly summaries.
- A bus allocation suggestion system based on previous booking and allocated bus data.

With these features, CampusWay offers a smart, reliable, and practical solution for university transportation issues and provides a smooth experience for both students and administration.

II. Literature Review

University transportation management has faced efficiency challenges for a long time. During the 1990s, most campuses used manual methods or basic desktop databases, which were moved to errors, overbooking, and limited transparency. The 2000s brought the emergence of web-based booking platforms, reducing paperwork but still lacking real-time updates and analytics. By the 2010s and 2020s, mobile-friendly, combined with data-driven systems and e-ticketing improved operational efficiency and user experience.

These systems have been guided by a number of usability and software development frameworks. Incremental models give guarantees to build systems in successive modules while breaking down designs through prototypes, ensuring better usability and fewer errors.The goal of Human-Computer Interaction(HCI) principles is to design user-friendly interfaces that minimize overload and facilitate system use for students and administrators. The problems such as overbooking, lack of real-time updates, limited reports, and inefficient scheduling based on previous studies on campus transportation systems are highlighted. Some of these problems were resolved by web-based and mobile solutions, but they often lacked full integration and advanced analytics.

These frameworks and earlier research are incorporated into CampusWay in multiple ways. It uses incremental development with prototyping to implement modules like login, booking and admin dashboard. A user-friendly interface for students and administrators is guaranteed by HCI principles. The calculation module extends historical booking data to recommend bus allocations, extending previous work on predictive analytics.

Even with these developments, many existing systems still fail to offer complete reporting, flexible features or optimized resource allocation. Students frequently experience seat unavailability or missing booking history, while administrators face difficulties in scheduling and managing data effectively. These limitations highlight the need for a more efficient, real-time, and user-friendly platform.

In order to fill these gaps, this study introduces CampusWay, a web-based university bus management system designed to digitalize and automate transportation services. The platform offers real-time seat booking, comprehensive reporting and centralized administration, aiming to enhance both student convenience and administrative efficiency.

III. Methodology

This section describes the method, diagrams and implementation systems used to develop the CampusWay.

A.Software Development Methodology

CampusWay was developed using an Incremental Development Model, allowing the system to be built in successive modules with iterative testing and refinement. Each module was implemented and tested before integrating with others or the next part , ensuring a stable and user-friendly system. . Depending on the project's size and complexity, it has several stages, as shown in figure-1. Generally each increment follows these stages:

I. Communication

For each increment, students and administrators were consulted to identify functional and non-functional requirements. For example, in the Seat Booking increment, requirements included real-time seat availability, booking history and cancellation options.

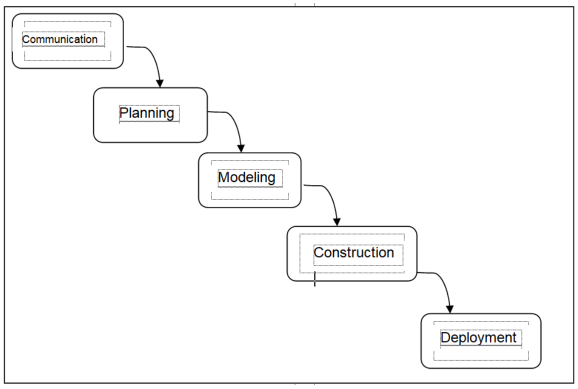


Figure 1: Incremental Software Development Model

II. Planning

For every increment, tasks, resources, and deadlines were established. The scope of the increment was decided, ensuring that modules such as Login & Registration, Seat Booking, Admin Dashboard, and Driver Module were developed sequentially with proper prioritization.

III. Modeling

Figma was used to create prototypes in order to visualize user interfaces and collect feedback. To ensure a clear design prior to coding, supporting diagrams such as the Use Case, Class Diagram and DFD Diagram were created for every increment.

IV. Construction

Each module was implemented using Laravel(PHP), MySQL, HTML and CSS.

V. Deployment

Once an increment was developed and tested individually, it was integrated into the main system. System testing was done to verify module compatibility, data consistency and overall functionality.After that, the increment was made available for user review and input. During this phase, frequent updates and bug fixes were implemented.

B. System Design

I. Use Case Diagram

Use case diagrams are used to show the system’s high level design in relation to its primary users, which are the administrator and the students.

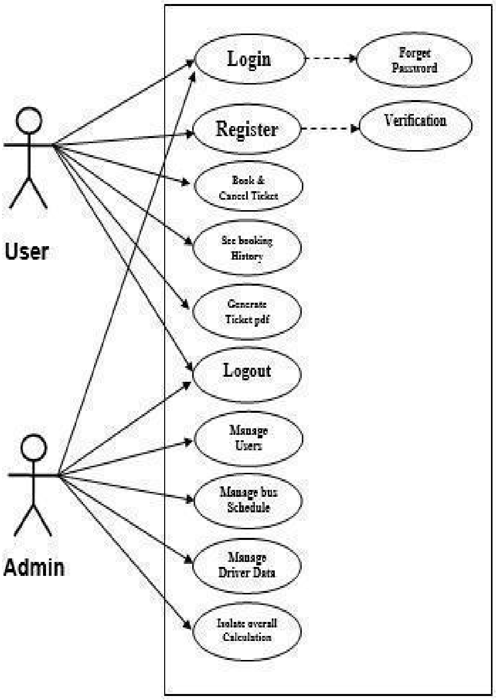


Figure 2: Use Case Diagram

Administrator Module:
It allows admin to manage the overall system by handling student accounts, buses and schedules. Administrators can monitor seat bookings in real time, generate reports for usage and system performance and system performance and update or remove bookings when necessary. They also ensure the availability of buses, modify schedules to meet student demand, which guarantees the CampusWay platform runs smoothly and is managed effectively.
Student Module:
It allows students to register and log in using their credentials, recover forgotten passwords and access the system to perform booking-related tasks. Students can search for available buses, check seat availability, and proceed with booking or canceling a seat according to their needs. They can also view their booking history and receive digital tickets via email, ensuring convenience and accessibility in managing their transportation needs.

II. DFD Diagram (Level-0 & Level-1)

The Data Flow Diagram of CampusWay shows how data flows between students, administrators, processes, and the database. Students can register, login, book or cancel tickets, and view account information like past bookings and upcoming rides at the context level. Once confirmed, the system generates e-tickets and sends them via email. Administrators manage bus schedules, routes, drivers, and buses, while also monitoring user activities, maintaining records, and generating system summaries. Every interaction goes via the CampusWay system, which interacts with the central database and handles requests.

In Level 1 DFD, registration and login validate student details with stored records before allowing access. Ticket booking checks seat availability, updates schedules, and confirms bookings, while cancellations restore seats and update records. By connecting buses, drivers, and routes, administrators can add, modify, and remove data from schedules. They also manage users, monitor activities, and generate reports from booking and cancellation data for future analysis. Overall, the DFD highlights that students can handle ticket booking and account management, whereas administrators focus on schedule management, user control, and performance reporting. The system ensures smooth data processing, accuracy, and transparency in managing university bus seat bookings.

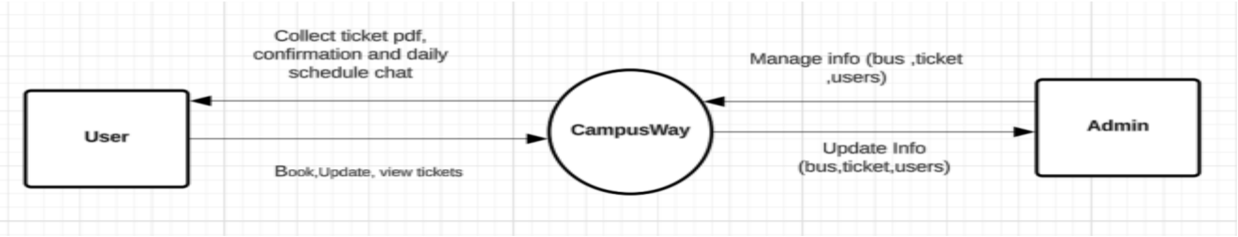

Collect ticket pdf, confirmation and daily schedule chat
User
Book,Update, view tickets
CampusWay
Manage info (bus ,ticket ,users)
Admin
Update Info (bus,ticket,users)

Figure 3: DFD Diagram (Level-0)

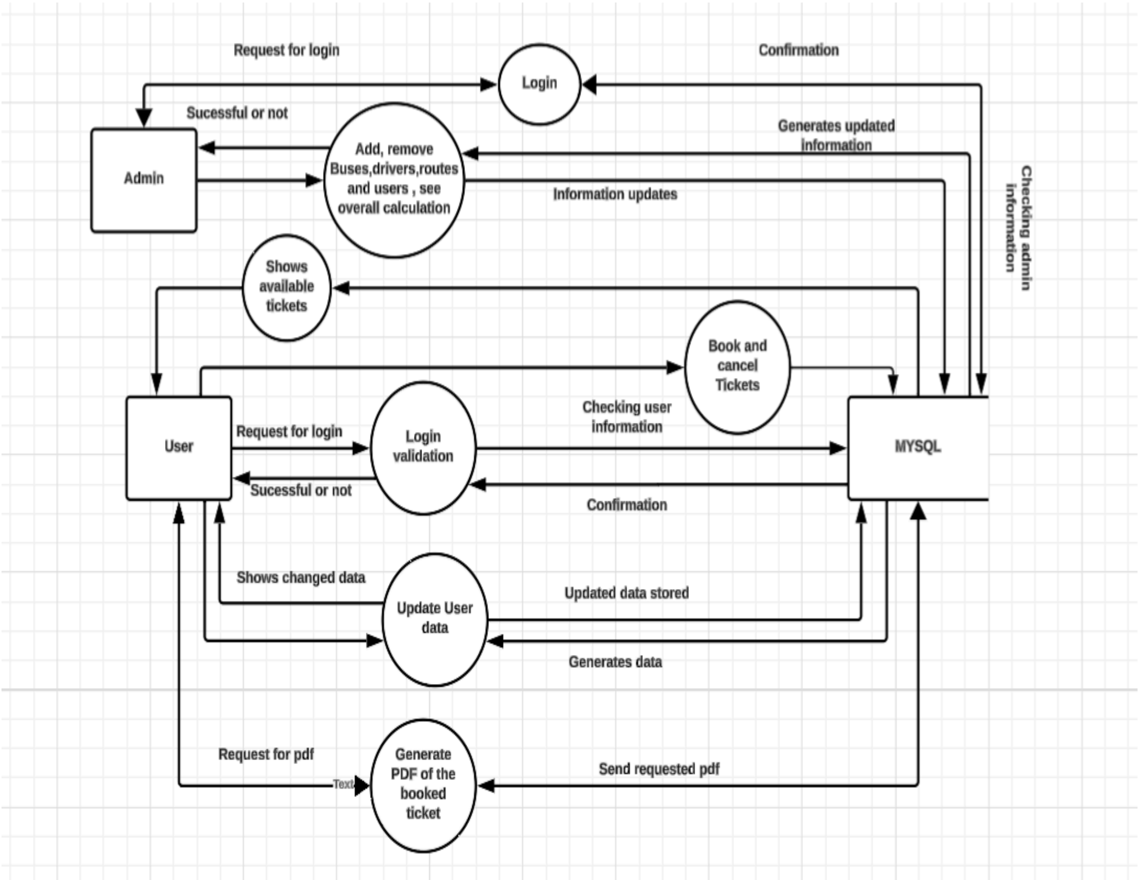


Figure 4: DFD Diagram (Level-1)

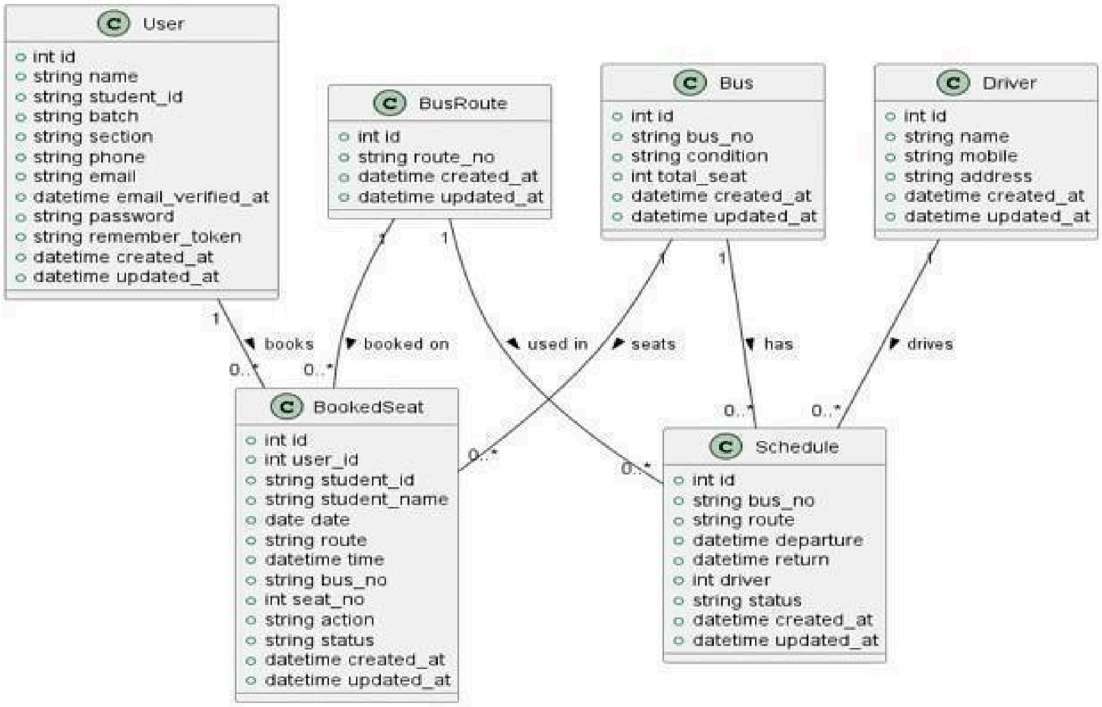


Figure 5: Class Diagram

III. Class Diagram

The UML class diagram represents the key structures of the bus ticket booking system. It mainly focuses on the crucial classes, their properties and relationships among them. This structure ensures the system functions effectively..

1. *User Class*
The user class depicts the individuals utilizing the system primarily student reserving seats. It stores fundamental information about students such as student_id, email, password in addition-timestamps (created_at, updated_at) for monitoring changes and updates. A single user is able to reserve one seat at a given time and also for return from university.

2. *Driver Class*
The driver class stores information about the bus drivers, including their contact details and address. Each driver can have multiple schedules to operate the bus service, creating a one-to-many relationship with the Schedule class.
3. *Bus Class*
The Bus class defines the physical buses available for booking. Attributes such as bus_no, condition, and total_seat describe the vehicle's identity and capacity. A bus can be associated with multiple schedules and multiple booked seats, resulting in two distinct one-to-many relationships with the Schedule and BookedSeat classes respectively.

4. *BusRoute Class*
The BusRoute class represents the predefined routes for bus services. Each route has a unique identifying number named route_no. Routes are connected to multiple schedules and booked seats that follow one-to-many relationships.

5. *Schedule Class*
The Schedule class stores the operational time table of buses. It includes departure time and return time, assigned drivers and the route the bus should service. The class holds foreign key references to Bus, Driver and BusRoute class that form many-to-many relationships.

6. *BookedSeat Class*

6. The BookedSeat class shows the records of individual seat reservations made by users. It contains booking meta data such as date, time, seat_no, status. Each booked seat linked with one user, one bus and one route. This forms a many-to-one relationship with User, Bus and BusRoute classes. This is the main class for the booking functionality system.

C. Tools and Instruments

Development Tools: Laravel, MySQL, HTML, CSS
Prototyping Tools: Figma
Server Environment: Local XAMPP server

D. System Validation and Readability

The validity and reliability of the CampusWay system were ensured by cross-checking system-outputs, verifying database updates for reservations and cancellations, and testing each module with real users. Detailed documentation of development stages, database schema, and functional specifications, ensuring that the system is accurate, user-friendly, and can be systematically reproduced or extended for future enhancements.

IV. Result Analysis and Discussion

This website successfully managed student registrations, real-time seat bookings, cancellations and administrative operations. The student module provided easy access to booking history and upcoming tickets, while the admin module efficiently handled buses, schedules and user control. Data gathered during testing showed better bus allocation, reduced manual effort, and increased operational efficiency. The calculation module enabled weekly, monthly and semester-wise summaries, supporting data-driven decision-making.

A. Registration/Login Module

Any user without his own student ID and password cannot access the system and has options to register as a student in the registration page. However, the user that has registered before can login using the Login page shown in the Figure. Also there is a forget password option through which a user can recover their password through email.

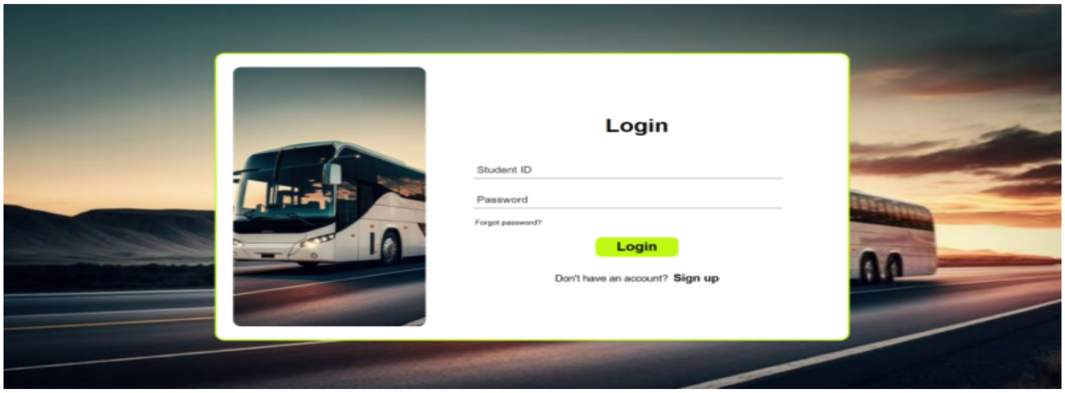
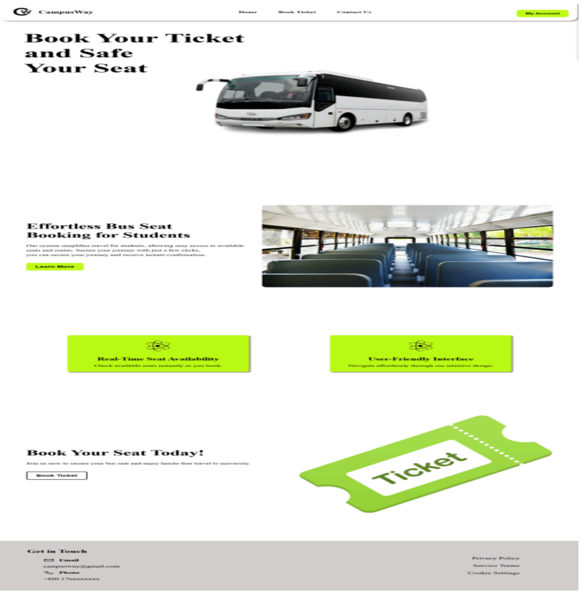


Figure 6: Login Page

B. Landing Page Module

The landing page of CampusWay works as the first center interactive point for users and changes dynamically based on their login status. Before logging in, users see Home1, which shows the navigation bar on the top with Home, Book Ticket, and Contact Us options, along with Login and Sign Up buttons at the top right corner. Users cannot access the booking page until they log in, ensuring secure and authenticated operations.
After logging in, the page transitions to Home 2, but it is almost similar to Home1 so that users thought it was the same interface. However, the Login and Sign Up buttons are replaced with a My Account section on the same top right corner. In My Account, a sidebar displays the user's name and email, along with three options for History, Upcoming, and Logout. The History section shows previously traveled tickets with dates, while the Upcoming section lists future bookings and allows ticket cancellations. Cancelled tickets are automatically restored as available in the booking page, maintaining real-time seat availability.



CampusWay Home Book Ticket Contact Us My Account

Book Your Ticket and Safe Your Seat

Effortless Bus Seat Booking for Students

Our system simplifies travel for students, allowing easy access to available seats and routes. Secure your journey with just a few clicks. you can secure your journey and receive instant confirmation.

Learn More

Real-Time Seat Availability

Check available seats instantly as you book.

User-Friendly Interface

Navigate effortlessly through our intuitive design.

Book Your Seat Today!

Join us now to secure your bus seat and enjoy hassle-free travel to university.

Book Ticket

Get in Touch

Email
campusway@gmail.com

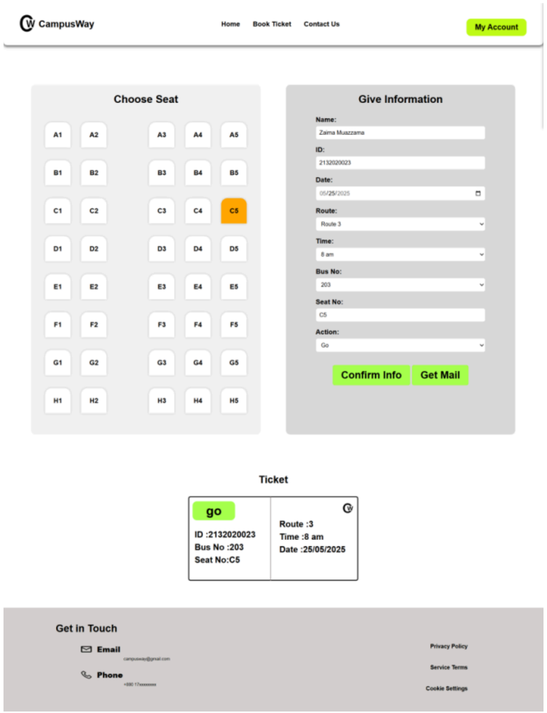
Phone
+880 17xxxxxxxx

Privacy Policy
Service Terms
Cookie Settings

Figure 7: Landing Page

C. User Booking Module

Users can switch to this page from home 2 by using navigation bars book ticket section. When we enter this page we can see a grid layout on the left side of the page which contains ticket layout, after selecting the date and bus number. There is an option of standing ticket also shown in layouts middle part. On the right side there is a form where users should fill up their name, student id, date, route, time, bus no, seat no, action(return/go). After fill-up this form and selecting the seat there are two buttons under the form confirm-info and get-mail. By clicking confirm info the page auto scrolled and generated an online ticket and if we click into the get mail button the PDF version of the ticket sent to the registered mail of the user. Users can download this PDF.



CampusWay Home Book Ticket Contact Us My Account

Choose Seat

A1	A2		A3	A4	A5
B1	B2		B3	B4	B5
C1	C2		C3	C4	C5
D1	D2		D3	D4	D5
E1	E2		E3	E4	E5
F1	F2		F3	F4	F5
G1	G2		G3	G4	G5
H1	H2		H3	H4	H5

Give Information

Name: Zaima Muazzama
ID: 2132020023
Date: 05/25/2025
Route: Route 3
Time: 8 am
Bus No: 203
Seat No: C5
Action: Go

Confirm Info Get Mail

Ticket

go
ID :2132020023
Bus No :203
Seat No:C5

Route :3
Time :8 am
Date :25/05/2025

Get in Touch

Email
campusway@gmail.com

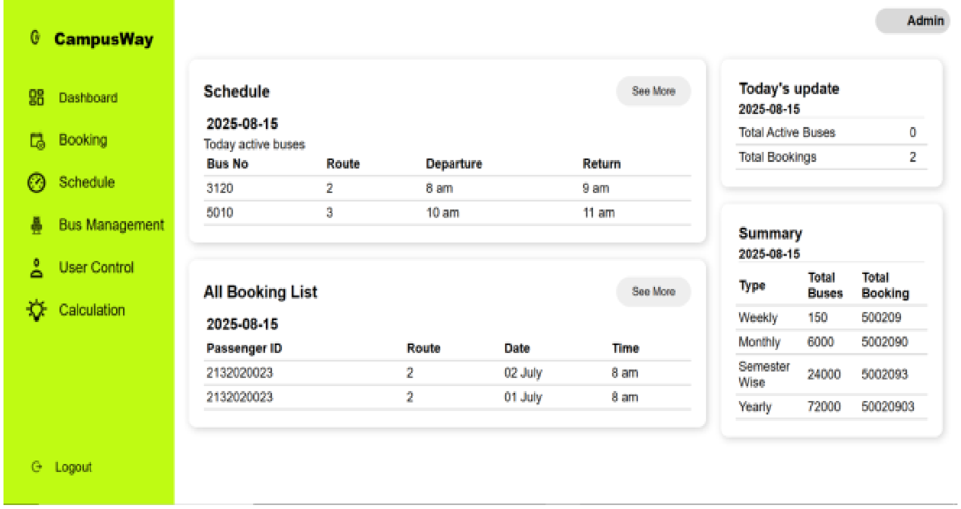
Phone
+880 17xxxxxxxx

Privacy Policy
Service Terms
Cookie Settings

Figure 8: Booking Page

D. Administration Module

There are multiple connected sections that make up the admin panel. The administrator has the ability to add, modify, and remove buses, routes, and drivers in the bus management section. This information is then utilized in the schedule section to prevent duplicate entries. By combining bus, route, driver, and time, the administrator can create daily schedules in the schedule section. This will automatically update the schedule table, database, and user dropdowns on the booking page. If required, the administrator can view and remove registered users from the user control section. Along with the daily number of buses in operation, the calculation section automatically creates summaries on a weekly, annual, and semester by user using basis. This data can also be utilized as real time information for future students.



Schedule

2025-08-15
Today active buses

Bus No	Route	Departure	Return
3120	2	8 am	9 am
5010	3	10 am	11 am

All Booking List

2025-08-15

Passenger ID	Route	Date	Time
2132020023	2	02 July	8 am
2132020023	2	01 July	8 am

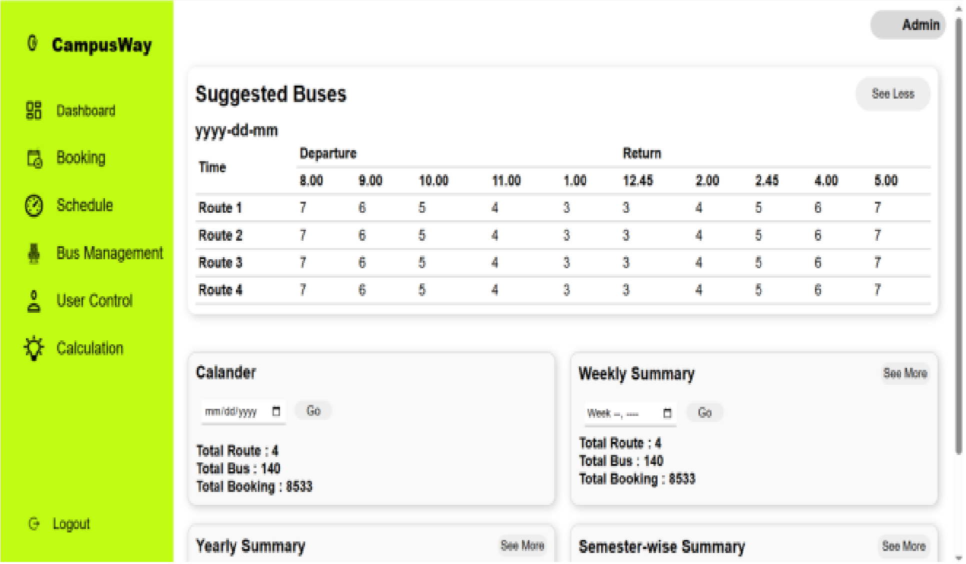
Today's update
2025-08-15

Total Active Buses	0
Total Bookings	2

Summary
2025-08-15

Type	Total Buses	Total Booking
Weekly	150	500209
Monthly	6000	5002090
Semester Wise	24000	5002093
Yearly	72000	50020903

Figure 9: Admin Dashboard



Suggested Buses

yyyy-dd-mm

Time	Departure					Return				
	8.00	9.00	10.00	11.00	1.00	12.45	2.00	2.45	4.00	5.00
Route 1	7	6	5	4	3	3	4	5	6	7
Route 2	7	6	5	4	3	3	4	5	6	7
Route 3	7	6	5	4	3	3	4	5	6	7
Route 4	7	6	5	4	3	3	4	5	6	7

Calander

mm/dd/yyyy

Total Route : 4
Total Bus : 140
Total Booking : 8533

Weekly Summary

Week --, ----

Total Route : 4
Total Bus : 140
Total Booking : 8533

Yearly Summary

Semester-wise Summary

Figure 10: Admin Calculation

V. Limitations and Future Work

Limitations
Our predictive transport system still bumps into everyday bumps. Sudden weather shifts- rain or extreme heat rarely show up in our booking logs, so forecasts crash when conditions change. Weekend festivals and exam-week rushes aren't flagged in the data, producing surprise spikes that our models miss. New or altered routes have zero history, leaving ML algorithms clueless until they gather enough bookings. Cancelled or duplicate entries introduce noise, driving models toward overfitting. Inference requests during peak hours can lag, undermining "real-time" claims. And stripping out student identifiers to protect privacy sometimes removes useful segmentation signals, making it harder to tailor predictions for different groups.

Future Work :
As our bus ticket booking system matures into a reliable platform for students and the university transport authority, the next logical advancement is the incorporation of machine learning (ML) to transform historical booking data into real-time, predictive insights. In the coming phases of this project, we will develop and deploy an end-to-end ML pipeline that anticipates demand fluctuations, optimizes bus allocations, and provides actionable recommendations to minimize overcrowding and resource under-utilization.First, we will formalize our research objectives. Forecast daily and route-specific seat demand with high accuracy. Recommend dynamic bus deployment strategies to balance capacity against forecasted demand. Detect unusual booking patterns that may indicate special events or operational disruptions. Integrate these predictions into the transport authority's decision-making dashboard in real time. By focusing on these goals, our future work will equip the transport authority with a proactive toolset, shifting operational planning from reactive adjustments to anticipatory scheduling.Handle missing or anomalous entries (e.g., canceled bookings) via median imputation or model-based estimation. Normalize numeric features (e.g., booking counts, capacity ratios) using min-max scaling to facilitate model convergence.Split the enriched data into training, validation, and test sets according to a time-aware scheme, ensuring that future dates are never used to train models for past predictions. Reserve the most recent month of data as a blind test set to evaluate real-world forecasting performance. By meticulously preparing our dataset in this manner, we will create a robust foundation for reliable and generalized ML models. By implementing this ML-driven enhancement, we expect to deliver measurable improvements. We anticipate at least a 30% decline in standing passengers on peak routes, as bus allocations will align more closely with forecasted demand. Avoiding underutilized trips will yield fuel and maintenance savings, improving the transport authority's budget efficiency. Reliable seat availability and fewer last-minute seat shortages will boost student satisfaction and reduce need for alternative transport. The authority will gain insights into long-term trends - such as semester-start spikes or exam-week surges - informing strategic route planning and permanent resource investments. These outcomes will collectively validate the value of integrating ML into university transport operations and establish a scalable blueprint for similar institutions.

VI. Conclusion

This study shows that moving from traditional methods to a structured digital platform can greatly improve how universities manage transportation. The proposed CampusWay system addresses the weaknesses of manual and semi-digital approaches by offering students features like real-time seat booking, cancellations, and booking history, while also giving administrators access to scheduling and reporting tools in one place. The study achieved its objectives by developing a dual-interface platform that integrates advanced features such as weekly, monthly, semester-wise, and yearly reporting, with real-time management of buses and routes. This ensures both student convenience and administrative efficiency. One of the most important contributions of this work is its integration of operational features with data-driven analytics. The calculation module, which recommends bus allocations per route based on historical booking data, marks a step toward intelligent, demand-driven resource management in campus transportation. This identifies the major gap in existing systems, which often lack adaptive and predictive capabilities. Future work will focus on extending the calculation module, which will give predictions for sufficient bus allocation by predictive analytics and machine learning. These developments will further optimize resource allocation and a better experience for the user.

References

[1]Obetta, J., Nwakwusi, L., & Onyiyne, O. (2023, February). Design and implementation of web-based smart campus bus transport management system with real-time display of arrival time. *ResearchGate*. https://www.researchgate.net/publication/380368080_Design_and_Implementation_of_web-based_Smart_Campus_Bus_Transport_Management_System_with_Real-Time_Display_of_Arrival_Time

[2]Md Noor, R., Rasyidi, N. B. G., Nandy, T., & Kolandaisamy, R. (2021). Campus shuttle bus route optimization using machine learning predictive analysis: A case study. *Sustainability*, 13(1), 225. https://doi.org/10.3390/su13010225

[3]Ganesh, K., Thrivikraman, M., Kuri, J., Dagale, H., Sudhakar, G., & Sanyal, S. (2012, June). Implementation of a real-time passenger information system. *arXiv preprint arXiv:1206.0447*. https://arxiv.org/abs/1206.0447

[4]Chen, G., Fei, X., Jia, H., Yu, X., & Shen, S. (2020, October). An optimization-and-simulation framework for redesigning university campus bus system with social distancing. *arXiv preprint arXiv:2010.10630*. https://arxiv.org/abs/2010.10630

[5]Jiang, F. (2020, October 15). Bus transit time prediction using GPS data with artificial neural networks. *Carnegie Mellon University*. https://www.ml.cmu.edu/research/dap-papers/F17/dap-jiang-fan.pdf(OUCI)

[6]Jindal, I., Chen, X., Nokleby, M., & Ye, J. (2017). A unified neural network approach for estimating travel time and distance for a taxi trip. *arXiv*. https://arxiv.org/abs/1704.02488(OUCI)

[7]Junyou, Z. (n.d.). Application of support vector machine in bus travel time prediction. *International Journal of Systems Engineering*, 2, 21.(OUCI)

[8]Mingheng, L., & Yujie, L. (2019). Bus arrival time prediction using support vector regression. *Journal of Transportation Engineering*, 145(4), 04019011. https://doi.org/10.1061/(ASCE)TE.1943-5436.0000705(MDPI)

[9]Gongyu, C., Xinyu, F., Huiwen, J., Xian, Y., & Siqian, S. (2020, October 20). An optimization-and-simulation framework for redesigning university campus bus system with social distancing. *arXiv preprint arXiv:2010.10630*. https://arxiv.org/abs/2010.10630(arXiv)

[10]Hiltgen, A., Kramp, T., & Weigold, T. (2006). Secure internet banking authentication. *IEEE Security & Privacy*, 4(2), 21-29. https://doi.org/10.1109/MSP.2006.32(arXiv)

[11]Baccelli, E., Hahm, O., Gunes, M., Wahlisch, M., & Schmidt, T. C. (2012). RIOT OS: Towards an OS for the Internet of Things. In *IEEE International Conference on Green Computing and Communications (GreenCom)* (pp. 187–194). IEEE. https://doi.org/10.1109/GreenCom.2012.58(arXiv)

[12]Currie, C. S. M., Fowler, J. W., Kotiadis, K., Monks, T., Onggo, B. K., Robertson, D. A., & Tako, A. A. (2022). A framework for operations research model use in resilience to climate change. *Environmental Modelling & Software*, 153, 105406. https://doi.org/10.1016/j.envsoft.2022.105406(arXiv)

[13]Cornell University Office of the University Registrar. (2021). The guide to Spring 2021 enrollment. https://registrar.cornell.edu/sites/registrar/files/Related-files/Guide%20to%20Spring%20Enrollment%2020201201.pdf(arXiv)

[14]Ganesh, K., Thrivikraman, M., Kuri, J., Dagale, H., Sudhakar, G., & Sanyal, S. (2012). Implementation of a real-time passenger information system. *arXiv preprint arXiv:1206.0447*. https://arxiv.org/abs/1206.0447
[15]Hiltgen, A., Kramp, T., & Weigold, T. (2006). Secure internet banking authentication. *IEEE Security & Privacy*, 4(2), 21-29. https://doi.org/10.1109/MSP.2006.32(arXiv)

[16]Baccelli, E., Hahm, O., Gunes, M., Wahlisch, M., & Schmidt, T. C. (2012). RIOT OS: Towards an OS for the Internet of Things. In *IEEE International Conference on Green Computing and Communications (GreenCom)* (pp. 187–194). IEEE. https://doi.org/10.1109/GreenCom.2012.58(arXiv)

[17]Currie, C. S. M., Fowler, J. W., Kotiadis, K., Monks, T., Onggo, B. K., Robertson, D. A., & Tako, A. A. (2022). A framework for operations research model use in resilience to climate change. *Environmental Modelling & Software*, 153, 105406. https://doi.org/10.1016/j.envsoft.2022.105406(arXiv)

[18]Cornell University Office of the University Registrar. (2021). The guide to Spring 2021 enrollment. https://registrar.cornell.edu/sites/registrar/files/Related-files/Guide%20to%20Spring%20Enrollment%2020201201.pdf

[19]Ganesh, K., Thrivikraman, M., Kuri, J., Dagale, H., Sudhakar, G., & Sanyal, S. (2012). Implementation of a real-time passenger information system. *arXiv preprint arXiv:1206.0447*. https://arxiv.org/abs/1206.0447

[20]Hiltgen, A., Kramp, T., & Weigold, T. (2006). Secure internet banking authentication. *IEEE Security & Privacy*, 4(2), 21-29. https://doi.org/10.1109/MSP.2006.32

[21]Baccelli, E., Hahm, O., Gunes, M., Wahlisch, M., & Schmidt, T. C. (2012). RIOT OS: Towards an OS for the Internet of Things. In *IEEE International Conference on Green Computing and Communications (GreenCom)* (pp. 187–194). IEEE. https://doi.org/10.1109/GreenCom.2012.58

[22]Currie, C. S. M., Fowler, J. W., Kotiadis, K., Monks, T., Onggo, B. K., Robertson, D. A., & Tako, A. A. (2022). A framework for operations research model use in resilience to climate change. *Environmental Modelling & Software*, 153, 105406. https://doi.org/10.1016/j.envsoft.2022.105406

[23]Cornell University Office of the University Registrar. (2021). The guide to Spring 2021 enrollment. https://registrar.cornell.edu/sites/registrar/files/Related-files/Guide%20to%20Spring%20Enrollment%2020201201.pdf

[24]Ganesh, K., Thrivikraman, M., Kuri, J., Dagale, H., Sudhakar, G., & Sanyal, S. (2012). Implementation of a real-time passenger information system. *arXiv preprint arXiv:1206.0447*. https://arxiv.org/abs/1206.0447
[25]Hiltgen, A., Kramp, T., & Weigold, T. (2006). Secure internet banking authentication. *IEEE Security & Privacy*, 4(2), 21-29. https://doi.org/10.1109/MSP.2006.32(arXiv)

[26]Baccelli, E., Hahm, O., Gunes, M., Wahlisch, M., & Schmidt, T. C. (2012). RIOT OS: Towards an OS for the Internet of Things. In *IEEE International Conference on Green Computing and Communications (GreenCom)* (pp. 187–194). IEEE. https://doi.org/10.1109/GreenCom.2012.58(arXiv)

[27]Currie, C. S. M., Fowler, J. W., Kotiadis, K., Monks, T., Onggo, B. K., Robertson, D. A., & Tako, A. A. (2022). A framework for operations research model use in resilience to climate change. *Environmental Modelling & Software*, 153, 105406. https://doi.org/10.101