Campusride: Smart College Bus Tracking and Transport Management System

Priyanka M[1], Sangeetha M[2]

[1],[2] Department of Computer Science and Engineering,

Adhiyamaan College of Engineering (Autonomous), Hosur, Tamil Nadu, India

Mentor: **Mrs. T. Vinothini**

Abstract—College transportation involves coordination among students, parents, drivers, and administrators. Traditional processes that depend on physical passes and manually maintained records can make it difficult to access transportation information efficiently. This paper presents CampusRide, a web-based college bus transportation management prototype that organizes transportation functions through role-specific interfaces for students, parents, drivers, and administrators. The application is developed using HTML, CSS, and JavaScript. It supports user registration and role-based login in the prototype, dashboard navigation, browser-based storage of user and transportation records, and digital bus-pass presentation using QRCode.js. The QR code represents a pass identifier or selected pass data; camera-based scanning and server-side validation are separate functions and are not claimed as completed in the current prototype. A bus-tracking interface is included, while continuous GPS transmission, route-based estimated arrival time, centralized multi-user storage, and automated push notifications require further integration. The paper describes the system architecture, modules, implementation approach, and prototype-level outcomes, and identifies the work needed to evolve the application into a secure real-time transportation platform.

Keywords—College transportation, bus management, QR code, web application, role-based access, digital bus pass, GPS tracking

1. INTRODUCTION

College buses are an important transportation facility for students travelling between their homes and an educational institution. Managing this service requires the maintenance of student, parent, driver, bus, route, and pass information. When these activities are handled through separate registers, printed passes, or telephone communication, users may find it difficult to obtain updated information and administrators may need to repeat manual record-management tasks.

CampusRide is designed as a web-based system that brings key college transportation functions into a single interface. It provides separate access areas for students, parents, drivers, and administrators. The prototype focuses on user registration and login, dashboard navigation, transportation information, bus fee and attendance views, and digital bus-pass QR generation. It also provides a bus-tracking interface that can serve as the entry point for future GPS integration.

The main contribution of this work is the design and implementation of a modular browser-based prototype that combines role-oriented transportation screens with digital QR bus-pass generation. The paper distinguishes currently demonstrated prototype functions from features that require external services or further implementation.

2. RELATED TECHNOLOGIES

GPS-based vehicle tracking uses location information from a GPS-enabled device or tracking unit. When location updates are transmitted to an application and displayed on a digital map, authorized users can observe the approximate vehicle position. Estimated arrival time additionally depends on route data, update frequency, and travel-time calculations.

QR codes are two-dimensional machine-readable symbols that can encode text, identifiers, or URLs. In a digital pass workflow, a unique pass identifier can be encoded and displayed. The QR image alone does not establish that a pass is genuine or active; a scanner must extract the identifier and a validation component must compare it with authorized records.

Web applications commonly use HTML to define page structure, CSS to control presentation, and JavaScript to implement client-side interactions. Browser Local Storage can retain key-value data on the same browser and is useful for demonstrations. It is not a centralized database and does not provide

secure multi-device synchronization by itself.

The reviewed project material identifies GPS tracking, QR-based identification, web-based management, and centralized digital records as relevant technology areas. CampusRide combines these concepts at the prototype interface level and identifies secure database, scanner, mapping, and notification integrations as further work.

3. PROPOSED SYSTEM

CampusRide provides four user categories: Student, Parent, Driver, and Admin. A user selects the appropriate role and accesses the corresponding interface after entering login details. The dashboard provides navigation to the available modules. The student interface presents student transportation details and links to bus-pass, attendance, fee, and tracking views. The parent interface presents related student and driver information. The driver interface provides access to assigned bus information and transportation functions. The administrator interface is intended for centralized management of users, buses, routes, and pass records.

The proposed system aims to reduce dependence on physical bus passes, make transportation information easier to access, and organize user-specific functions. Access to records should be restricted according to user roles in a production deployment. Parent-to-student account association and administrator authorization require trusted backend validation.

The system is designed to be extendable. A GPS-enabled mobile device can provide bus coordinates; a mapping service can display the coordinates and route. A notification service can deliver delay or arrival updates. These integrations are not assumed to be operational merely because a tracking or notification screen is present.

4. SYSTEM ARCHITECTURE AND METHODOLOGY

The architecture is organized into four logical layers. The presentation layer contains the browser interface created with HTML and CSS. The application-logic layer uses JavaScript to process user actions, validate form inputs, manage navigation, and prepare QR data. The data layer uses browser Local Storage in the current prototype to retain user, session, and selected transportation information. The external-services layer represents optional integrations such as GPS, digital maps, camera-based QR scanning, centralized databases, and push notifications.

During registration, the application collects the user's name, username, email address, password, and role. Basic client-side checks are applied before a user record is saved. During login, the entered credentials and selected role are compared with the records available in browser storage. A successful match opens the relevant module; otherwise, an error message is displayed. The prototype also initializes a demonstration administrator record.

For QR generation, the application obtains the selected pass data or identifier and passes it to the QRCode.js library. The generated QR image is displayed in the digital bus-pass interface. In a complete deployment, a scanner must read the encoded value and a separate validation service must check pass status, expiry, and the associated student record.

For live tracking, a GPS-enabled device must periodically transmit location data to a backend or tracking service. The application can then retrieve the latest authorized location and display it on a map. Route-based ETA calculation and automated notifications require additional data sources and services.

5. IMPLEMENTATION

The prototype is implemented with HTML5, CSS3, and JavaScript and is edited using Visual Studio Code. It can be executed in a compatible web browser, such as Google Chrome or Microsoft Edge. QRCode.js is used for QR image generation. Browser Local Storage is used for prototype persistence through application-specific database and session keys.

The Student Module provides access to student details, bus pass, attendance, bus fee, and tracking views. The Parent Module presents student and driver details and provides navigation to attendance and bus-related information. The Driver Module presents assigned bus information and provides access to attendance, tracking, and bus-pass-related screens. The Admin Module is designed to support management of transportation records. The exact operations available depend on the functions implemented in the prototype.

The implementation uses client-side role checks and browser storage for demonstration. This approach is suitable for a local prototype but should not be treated as production-grade authentication or secure storage. Passwords should not be stored in plain text in a production system. A trusted backend, secure

password hashing, authorization rules, and a centralized database are required before deployment for real student data.

6. RESULTS AND DISCUSSION

The project report records a browser-based dashboard and separate Student, Parent, and Driver screens. It also describes registration and login, dashboard navigation, local storage of user and transportation-related records, and QR-code generation for registered users. The student screen provides access to bus-pass, attendance, bus-fee, and tracking views. The parent and driver screens organize the corresponding transportation information.

These outcomes demonstrate the prototype's user-interface organization and client-side workflow. They do not establish that continuous GPS tracking, automatic ETA calculation, camera-based QR verification, centralized multi-device synchronization, or push notifications are operational. Those functions depend on additional services and testing. No measured performance, accuracy, or user-study results are provided in the source report; therefore, quantitative improvement claims are not made in this paper.

The main current limitation is the use of browser Local Storage, which is restricted to the browser/device context. The QR library generates a code but does not perform scanning or validation. Similarly, a tracking interface alone does not provide live vehicle coordinates. These boundaries are important when interpreting the prototype results.

7. CONCLUSION AND FUTURE WORK

CampusRide presents a web-based prototype for organizing college transportation information through Student, Parent, Driver, and Admin modules. The application demonstrates the use of HTML, CSS, and JavaScript for an interactive interface, browser-based data persistence for demonstration, and QRCode.js for digital bus-pass QR generation. The modular design provides a foundation for further development of college transportation services.

Future work includes integrating a centralized database such as Firebase Firestore or MySQL, implementing secure authentication and role-based authorization, adding camera-based QR scanning and server-side pass validation, and connecting GPS devices to a digital map. Route-aware ETA calculation, push notifications, emergency alerts, analytics, and a dedicated mobile application can also be developed. Each feature should be implemented and tested before it is represented as operational.

REFERENCES

[1] "Optimizing Urban Public Transportation Through AI-Based Bus Tracking and QR Code Ticketing," IEEE ICICIS, 2025. (Bibliographic details as listed in the project report.)

[2] "EduBusGuru: Enhancing College Bus Tracking and Management with Real-Time Mobile App," IEEE ICCCNT, 2024. (Bibliographic details as listed in the project report.)

[3] "Estimation of Bus Arrival by Tracking and Analyzing the Users Real Time Location," IEEE ICSCAN, 2023. (Bibliographic details as listed in the project report.)