

CHAPTER – I

1. INTRODUCTION

1.1 PROJECT OVERVIEW

The primary objective of the QR Code and Website Bus Route Tracking Information System project is to revolutionize the public transportation experience by integrating cutting-edge technology into everyday travel. This system aims to enhance the efficiency, accessibility, and user-friendliness of public bus services through the innovative use of QR codes and an informative website. By streamlining the process of accessing bus route information, providing real-time updates on bus arrivals, and facilitating seamless communication between passengers and service providers, the project seeks to address common challenges faced by public transit users, including the unpredictability of bus schedules and the difficulty in accessing up-to-date route information.

Central to achieving this objective is the development of a user-centric platform that caters to the needs and preferences of public transit users. The system is designed to empower users with the tools and information necessary to make informed decisions about their travel plans, thereby reducing wait times, improving the predictability of journeys, and enhancing overall satisfaction with public transportation services. Through the use of QR codes, passengers can quickly access detailed information about bus routes, stops, and fares without navigating through multiple web pages, making the system both efficient and user-friendly.

The study also focuses on the role of administrators in maintaining the accuracy and reliability of the system. By providing administrators with a robust back-end platform for managing bus and stop details, the system ensures that users have access to current and comprehensive information. This aspect of the project highlights the importance of effective data management and the need for a responsive and adaptable system that can accommodate changes in bus schedules, routes, and fares.

An essential component of the project is the feedback and query submission feature, which establishes a direct line of communication between users and administrators. This module not only allows users to report issues or submit inquiries but also enables administrators to gather valuable insights into user experiences and expectations. By analyzing feedback and addressing queries promptly, the system aims to continuously improve and evolve in response to the needs of its users.

Moreover, the study explores the potential of QR technology and digital platforms to transform public transportation into a more accessible, reliable, and enjoyable service. By leveraging these technologies, the project seeks to bridge the gap between traditional transit systems and the digital age, offering a model for future innovations in public transportation management and user engagement.

The "QR Code and Website Bus Route Tracking Information System" represents a significant step forward in the quest to enhance public transportation services. Through the innovative use of technology, the system aims to improve the travel experience for users, streamline service management for administrators, and pave the way for a more efficient, accessible, and user-friendly public transportation system. By addressing the challenges and leveraging the opportunities presented by digital technologies, this study sets the foundation for transformative changes in how public transit services are accessed, managed, and experienced.

1.2 SYSTEM SPECIFICATION

1.2.1 HARDWARE SPECIFICATION

Processor	:	Intel Dual Core Processor
Ram	:	1 GB RAM
Hard Drive	:	160 GB
Monitor	:	17 INCHES
Keyboard	:	104 keys
Mouse	:	Logitech Optical Mouse

1.2.2 SOFTWARE SPECIFICATION

Operating System	:	Windows 10 onwards
Front-end	:	Microsoft ASP .Net 2013
Back-End	:	MS SQL SERVER 2012

CHAPTER – II

2. SYSTEM STUDY AND ANALYSIS

2.1 EXISTING SYSTEM

The current system for managing bus routes and providing information to users relies heavily on manual inputs and traditional communication methods, leading to inefficiencies and a lack of real-time updates. Information about bus routes, stops, and schedules is often outdated or inaccessible, causing inconvenience for passengers. The system lacks dynamic interaction with users, offering no direct way for feedback or queries, and does not utilize modern technologies like QR codes for easy access to route information.

2.1.1 SCOPE AND LIMITATIONS

1. Reliance on manual updates leads to outdated route and schedule information.
2. Lack of real-time tracking for buses limits the accuracy of arrival times.
3. No direct feedback mechanism for users to report issues or make inquiries.
4. Absence of a user-friendly interface for easy access to bus information.
5. Inadequate use of technology for efficient information dissemination.

2.2 PROPOSED SYSTEM

The proposed system introduces a comprehensive QR Code and Website Bus Route Tracking Information System aimed at enhancing the user experience through digital transformation. This system integrates QR code technology for instant access to up-to-date bus routes, schedules, and stop details. It offers real-time tracking of buses, allowing users to plan their journeys more effectively. By incorporating user registration and login modules, the system personalizes the experience and opens channels for direct feedback and queries, ensuring users' voices are heard and addressed promptly.

2.2.1 FEATURES

1. Real-time bus tracking enhances journey planning with up-to-date information.
2. QR codes at bus stops provide instant access to route details and schedules.
3. User feedback mechanism allows for direct communication and swift issue resolution.
4. Personalized user accounts for tailored information and notification preferences.
5. Comprehensive administrator portal for efficient management of bus details, stop information, and user queries.

CHAPTER – III

3.SYSTEM DESIGN AND DEVELOPMENT

3.1 FILE DESIGN

File design plays a crucial role in the development and efficiency of the QR Code and Website Bus Route Tracking Information System. This system requires a well-structured file design to manage and store data effectively, ensuring quick access, security, and integrity of the information. The file design encompasses various aspects, including database design, file storage for QR codes, user information storage, and logs for feedback and queries.

The heart of the system lies in its database, where critical information about bus routes, stops, schedules, and user details are stored. The database is structured into several key tables: BusDetails, StopMaster, UserAccounts, and FeedbackQueries. The BusDetails table stores information about each bus, including a unique bus number, starting point, and ending point. The StopMaster table holds details about each stop for every bus route, such as stop number, stop name, arrival time, and charges, enabling the system to provide detailed route information to the users.

QR codes represent an innovative approach to delivering bus route information efficiently. Each QR code generated contains embedded data about bus routes and stops, which is stored in a separate file system directory within the web application. The QR codes are linked to their respective bus numbers in the database, allowing for dynamic generation and updating as route information changes. This design ensures that users always have access to the most current information by scanning the QR codes available at bus stops or on the website.

User information is managed through the UserAccounts table, which captures details such as usernames, passwords (encrypted for security), email addresses, and user types (regular user or administrator). This segregation allows for personalized experiences for users, who can log in to access specific route information, track buses in real-time, and submit feedback or queries directly through the system.

Feedback and queries from users are a valuable source of information for continuous improvement. These are captured in the FeedbackQueries table, which stores details about the user's feedback, type of query, submission date, and status (resolved or unresolved). This

enables the administrators to monitor and address user concerns promptly, enhancing the overall service quality.

In addition to the primary data storage, the system incorporates logs for tracking user activities and system operations. These logs are essential for debugging, monitoring system usage patterns, and ensuring security. They record events such as login attempts, information access, QR code scans, and system errors, providing a comprehensive view of the system's performance and user interactions.

The file design for the QR Code and Website Bus Route Tracking Information System is built with scalability and flexibility in mind, allowing for future expansions such as adding new bus routes, stops, and user features. By employing a combination of relational database management for structured information and file storage for QR codes and logs, the system ensures efficient data management and accessibility. This design not only supports the current requirements of providing timely and accurate bus route information but also lays the foundation for integrating advanced features and enhancements to meet the evolving needs of users and administrators alike.

3.2 INPUT DESIGN

Input design is a critical phase in the development of the QR Code and Website Bus Route Tracking Information System, focusing on how users interact with the system and how the system captures, handles, and validates user input. This component is essential for ensuring user satisfaction, data integrity, and overall system effectiveness. The input design strategy for this project encompasses various interfaces and mechanisms for data entry, emphasizing ease of use, accuracy, and efficiency.

User Interface Design

The user interface (UI) is designed to be intuitive and user-friendly, catering to a diverse user base with varying levels of technical proficiency. The UI includes forms for user registration, login, bus route searches, stop details inquiries, QR code scans, and feedback submissions. Each form is structured to minimize user effort and guide them through the input process with clear instructions, input fields, drop-down menus, and validation messages.

- **User Registration and Login:** The registration form captures essential details such as username, email, password, and confirmation of password. Password fields employ masking to ensure privacy, and an email verification mechanism is implemented to validate

the user's email address. The login interface is straightforward, requiring only the username and password, with an option to remember the user for future visits.

- **Bus Route and Stop Details:** Users can find routes and stops by entering bus numbers or selecting stops from a dropdown menu populated dynamically based on the database. This approach reduces errors by limiting choices to valid entries. Auto-complete features further enhance the user experience by suggesting possible entries as the user types.
- **QR Code Generation and Scan:** Administrators can generate QR codes for bus routes through a simple interface where they select a bus number and click a button to create the QR code. Users scan QR codes using their mobile devices, which are then processed by the system to display relevant route information without requiring manual input.
- **Feedback and Queries:** A dedicated form allows users to submit feedback or queries, categorizing their submission for efficient processing. This form includes text areas for detailed descriptions and options to upload relevant attachments for clarity.

3.3 OUTPUT DESIGN

Output design is a fundamental aspect of the QR Code and Website Bus Route Tracking Information System, focusing on how the system presents information to users in a clear, accessible, and actionable manner. This phase is crucial for ensuring that users can easily understand and act upon the information provided by the system. The output design strategy encompasses various display formats, information architecture, and interaction design principles to enhance user engagement and satisfaction.

3.3.1 Information Display and Visualization

The system employs a variety of methods to display information effectively to users and administrators, including tables, lists, interactive maps, and QR codes. Each method is chosen based on the type of information being presented and the user's context.

- **Bus Route and Stop Details:** Information about bus routes, including stop names, arrival times, and charges, is presented in tabular format for easy comparison and understanding. The tables are sortable and filterable to allow users to customize the view according to their preferences.

- **QR Code Display:** QR codes generated for each bus route are displayed prominently for easy scanning. Additional details about the route are provided adjacent to the QR code to give users context before scanning.
- **Interactive Maps:** For users seeking bus route information, interactive maps display the route with markers for each stop. Users can click on markers to see detailed stop information, enhancing their planning process with a visual representation of the route.
- **Feedback and Query Responses:** Responses to user queries and feedback are formatted as personalized messages, ensuring clarity and demonstrating the system's attentiveness to user inputs. Administrators receive notifications and summaries of user interactions in dashboard widgets, allowing them to quickly assess and respond to user needs.

3.3.2 User Experience and Accessibility

Consistent with the input design, the output design adheres to accessibility standards to ensure that all users, including those with disabilities, can access and understand the system's outputs. This includes the use of alt text for images, accessible tables, and compliance with color contrast ratios for users with visual impairments.

1. **Responsive Design:** The system's output design is responsive, ensuring that information is presented effectively across devices of varying screen sizes, from desktops to mobile phones. This responsiveness enhances the usability of the system, particularly for users accessing QR codes and route information on the go.
2. **Real-time Updates:** For dynamic information such as bus arrival times, the system provides real-time updates. This ensures that users have access to the most current information, critical for planning their travel. Real-time data are presented in a way that highlights updates, such as flashing indicators or automated refreshes, to draw user attention.
3. **Customization and Personalization:** Where possible, the system offers users the ability to customize the output presentation, such as choosing which columns to display in tables or setting preferences for notification types. This personalization enhances user engagement by allowing users to tailor the system to their specific needs.

3.3.3 Interaction Design

The system's output design incorporates principles of interaction design to facilitate user engagement with the information presented. This includes clear call-to-action buttons,

interactive elements that provide additional details on hover or click, and intuitive navigation pathways that guide users through the information hierarchy.

- **Feedback Loops:** The design includes feedback loops, such as confirmation messages after successful actions (e.g., QR code scans, feedback submission) and error messages that are informative and suggest corrective actions. These feedback mechanisms ensure users are aware of the system state and can navigate the interface with confidence.
- **Guided Actions:** For complex outputs, such as route planning results, the system offers guided actions, suggesting the next steps users can take. For example, after scanning a QR code to view a bus route, users may be prompted to bookmark the route for future reference or share it with others.

3.4 DATABASE DESIGN

The development of a comprehensive database design is fundamental in ensuring the seamless operation of an application, particularly one that necessitates the management of complex data related to bus routes, user queries, and registration information. This document presents an elaborate design for a database system consisting of four main tables: BusDetails, QueryTab, StopMaster, and Userreg. Each of these tables serves a distinct purpose within the system, supporting various functionalities from tracking bus routes to managing user-generated queries and registrations.

BusDetails Table

The BusDetails table is a cornerstone of this database design, capturing essential information about each bus service offered. It uniquely identifies each bus by a BusNumber, which serves as the primary key. This table includes columns for Starting_point and ending_point, providing clear demarcations for the bus routes. This setup not only facilitates the tracking of bus services but also aids in planning by providing insight into the coverage area of each bus service.

QueryTab Table

User engagement and feedback are captured in the QueryTab table, which records queries or complaints submitted by users. The table is uniquely indexed by Query_no, ensuring each entry is distinct. It includes a foreign key User_id, which references the Userid in the Userreg table, establishing a direct relationship between each query and the user who posted

it. This structure enables a responsive system where user feedback is systematically recorded and associated with specific users for follow-up or statistical analysis.

StopMaster Table

The StopMaster table enriches the database by detailing each stop along a bus route. It is indexed by Stop_id as its primary key and includes a foreign key BusNumber that links back to the BusDetails table. This table not only lists the stops (identified by Stop_name) in the order they are encountered (Stop_no) but also provides additional data such as Arrival_time and Charges. This information is vital for route planning, fare calculation, and providing passengers with accurate, timely information about bus schedules.

Userreg Table

The Userreg table is designed to manage user registrations within the system. Each record in this table corresponds to a unique user, identified by Userid as the primary key. It stores personal information such as name_user, gender, email_id, and mobile_number, alongside credentials used for system access (username and password). This table is pivotal for implementing authentication mechanisms and personalizing user experiences based on the stored demographic information.

Database Relationships

The relational structure between these tables is meticulously designed to ensure data integrity and facilitate efficient data retrieval. For instance, the foreign key User_id in the QueryTab table and BusNumber in the StopMaster table create explicit connections between user queries, bus details, and bus stops. These relationships enable complex queries across the database, such as retrieving all queries made by a specific user or listing all stops for a particular bus route.

Normalization and Data Integrity

The database design adheres to normalization principles to eliminate redundancy, ensure data integrity, and optimize storage. By segregating information into dedicated tables and establishing clear relationships through primary and foreign keys, the design mitigates the risks of data anomalies. Constraints and validation rules can be implemented to further safeguard data integrity, ensuring that only valid data is entered into the system.

Scalability and Flexibility

This database design is not only robust but also scalable and flexible. It supports the addition of new bus routes, modifications to existing routes, and updates to user information with minimal disruption. The modular nature of the design allows for future expansions, such as introducing new tables to accommodate additional features like real-time tracking, loyalty programs, or expanded user profiles.

3.5 MODULE DESCRIPTION

1. Admin Login

Designed exclusively for system administrators, this module offers secured access to the backend of the system. Administrators can manage bus routes, stop details, and user queries, ensuring that the system remains up-to-date and functional.

2. Bus Details

Accessible to administrators, this module provides the capability to add, update, or remove bus routes and associated details. It serves as the central repository for all information related to bus services, ensuring that users have access to accurate and comprehensive data.

3. Stop Details

Similar to the Bus Details module but focused on bus stops, this feature allows administrators to manage stop-specific information such as location, scheduled arrival times, and any special notes or alerts relevant to passengers.

4. User Registration and Login

This module provides the initial interface for new users to register within the system. It captures essential user information such as name, email, and contact details, and facilitates the creation of a unique username and password for subsequent logins. The login functionality ensures that users can securely access their accounts, view personalized route information, and manage their queries or complaints. This module is crucial for personalizing the user experience and safeguarding user data.

5. Bus Route

A core functionality of the system, this module allows users to search for bus routes based on their starting point and destination. It provides comprehensive details, including all stops along the route, the fare for the journey, and estimated travel times. This module helps users plan their travel efficiently, offering real-time data to make informed decisions about their transit options.

6. Stop Details

Focused on providing timely information, this module lists the arrival times of buses at specific stops. Users can select a stop and view the schedule of all buses arriving at that location, helping them to better coordinate their travel plans. This module is integral to improving the predictability and reliability of bus services for passengers.

7. QR Scan

Leveraging the QRCoder.dll, this innovative module allows users to generate and scan QR codes associated with specific bus routes. Each QR code contains detailed information about the route, stops, and fare, accessible with a simple scan. This feature simplifies the access to route details, making it a convenient option for users on the go and enhances the digital interaction with the transit system.

8. Query

A platform for user feedback and engagement, this module enables users to post queries or complaints regarding bus services. It serves as a direct communication channel between passengers and transit authorities, facilitating timely responses and resolutions to any issues encountered. This module is key to maintaining high service standards and ensuring user satisfaction.

9. View Queries

This administrative module offers a consolidated view of all queries and complaints submitted by users. It enables administrators to respond to feedback, address concerns, and implement necessary changes or improvements based on user input.

10. View Registered Users

Providing insight into the user base, this module allows administrators to view the list of registered users, their profiles, and activity within the system. It is essential for managing user accounts and understanding user demographics for targeted service improvements.

CHAPTER – IV

4.SYSTEM TESTING AND IMPEMETATION

System testing is a crucial phase in the development of the QR Code and Website Bus Route Tracking Information System. It involves validating the integrated system to verify that it meets the specified requirements. This comprehensive testing phase encompasses various types of tests to ensure the system's functionality, reliability, performance, and security across all its components and user interfaces. Here, we will explore the different types of system testing executed for this project, highlighting their significance and outcomes.

4.1 UNIT TESTING

The first step in our testing process was unit testing, where individual units or components of the web application were tested in isolation. This was crucial for identifying any issues at the earliest stage of development. Developers wrote and executed test cases for discrete functions within each module, such as user registration, login functionality, and QR code generation. This ensured that each function performed as expected before integrating the units into the larger system.

4.2 INTEGRATION TESTING

After unit testing, integration testing was conducted to evaluate the combined parts of the application to determine if they worked together as intended. This step was essential for uncovering interface defects between integrated components. For example, testing the interaction between the 'Bus Route' module and 'Stop Details' module ensured that users could seamlessly retrieve bus stop information after selecting a specific route. Integration testing helped in identifying discrepancies in data flow and function calls between modules.

4.3 FUNCTIONAL TESTING

Functional testing focused on assessing each feature of the application against the functional requirements. Test cases were designed to cover all possible inputs and usage scenarios to ensure that the system behaved as expected. This included verifying the correctness of bus routes and stops displayed, accuracy of fare calculations, the functionality of the query system, and the integrity of the user registration and login processes. Special attention was

given to the QR code generation and scanning feature to ensure it accurately represented bus route details.

4.4 SYSTEM TESTING

With all components integrated, system testing was conducted to evaluate the complete and fully integrated application against the specified requirements. This end-to-end testing phase assessed the system's compliance, overall behavior, and user experience. Test scenarios included a mix of typical user interactions and edge cases, such as registering a new user, finding bus routes, scanning QR codes for route details, submitting queries, and admin functionalities like updating bus details and responding to user queries.

4.5 PERFORMANCE TESTING

Performance testing aimed to evaluate the system's stability, responsiveness, and scalability under various load conditions. Simulated users interacted with the application to measure response times, throughput rates, and the system's ability to handle concurrent access. This was particularly important for the 'Bus Route' and 'Stop Details' modules, where real-time information retrieval was critical. The system was tested for peak load performance to ensure that it could manage high user volumes without significant degradation in performance.

4.6 SECURITY TESTING

Given the nature of user data handled by the system, security testing was paramount. This involved assessing the application for vulnerabilities such as SQL injection, cross-site scripting (XSS), and ensuring that user data, especially passwords and personal information, were adequately protected. Authentication and authorization mechanisms were rigorously tested to prevent unauthorized access to sensitive admin functionalities and user information.

4.7 USABILITY TESTING

Usability testing involved real users interacting with the system to assess its ease of use, navigation, and overall user experience. Feedback was collected on the intuitiveness of the user interface, clarity of instructions, and visual design of the application. This feedback was instrumental in refining the UI/UX design, ensuring that users could easily register, log in, find bus routes, and utilize the QR code feature without confusion.

4.8 ACCEPTANCE TESTING

The final phase of testing was acceptance testing, where the completed application was presented to Users to ensure it met their expectations and requirements. This involved testing the application in a production-like environment to validate its readiness for deployment. User feedback was used to make final adjustments before the official launch.

CHAPTER – V

5. CONCLUSION

5.1 CONCLUSION

The QR Code and Website Bus Route Tracking Information System represents a significant leap forward in public transportation management and user engagement. This comprehensive system, designed with both the user and administrator in mind, not only streamlines the process of accessing and managing bus route information but also introduces innovative ways for users to interact with the service through QR codes. The culmination of this project reflects a deep understanding of the needs within the public transportation sector, focusing on accessibility, efficiency, and user satisfaction. The system's diverse modules, including User Registration and Login, Bus Route information, Stop Details, QR Scan, and Query functionalities for users, alongside the Admin Login, Bus Details, Stop Details, View Queries, and View Registered Users for administrators, demonstrate a holistic approach to addressing both ends of the spectrum. Users benefit from an intuitive platform that provides real-time data and a novel QR code feature for route information, enhancing their travel experience. Meanwhile, administrators are equipped with robust tools for managing route details, responding to user queries, and gaining insights into user engagement and feedback.

Throughout the development and deployment phases, the project underwent rigorous testing to ensure reliability, security, and usability. User feedback played a pivotal role in refining the system, highlighting the importance of a user-centric design approach. The successful integration of the `QRCode.dll` for the QR scan module exemplifies the project's innovative edge, offering a quick and convenient way for users to access detailed bus route information.

The project's implementation has not only met its initial objectives but has also set a new standard for digital solutions in public transportation. It showcases the potential of combining traditional service management with modern technology to create more accessible, efficient, and user-friendly transportation systems. The positive reception from both users and administrators alike underscores the system's effectiveness in enhancing the public transportation experience, contributing to increased user satisfaction and operational efficiency.

However, the journey does not end here. The system's architecture and foundational principles lay the groundwork for ongoing improvements and expansions. The feedback loop

established between users and administrators fosters a dynamic environment where the system can continuously evolve in response to changing needs and emerging technologies.

In conclusion, the QR Code and Website Bus Route Tracking Information System stands as a testament to the power of digital innovation in transforming public services. By bridging the gap between users and transportation services, the system not only improves the efficiency of bus route management but also enriches the user experience, setting a new benchmark for future developments in the field.

5.2 SCOPE FOR FUTURE ENHANCEMENT

1. Developing dedicated mobile apps for both iOS and Android platforms to enhance accessibility and user engagement.
2. Implementing GPS tracking for buses to provide users with live updates on bus locations and expected arrival times.
3. Incorporating voice command features to make the system more accessible, especially for users with visual impairments.
4. Adding multilingual support to cater to a diverse user base, improving inclusivity and accessibility.
5. Enhancing the query and feedback module to include rating systems and detailed reviews for specific routes or services.
6. Utilizing machine learning algorithms to analyze travel patterns and optimize bus routes and schedules.
7. Integrating payment gateways to enable users to purchase tickets or passes directly through the system.
8. Incorporating augmented reality features to assist users in navigating to bus stops or identifying bus routes.
9. Developing advanced analytics tools to provide administrators with deeper insights into user behavior, route popularity, and operational efficiency.
10. Adding features to promote sustainability, such as tracking carbon footprint savings from using public transport, to encourage eco-friendly travel habits.

BIBLIOGRAPHY

BOOKS REFERRED

1. Alex Homer , “**Professional ASP.NET 1.1**”, 2004 Edition, Wrox Publications
2. Steven Holzner, “**Visual Studio. NET Black Book**”, 2003 Edition, Dreamtech Publications
3. Roger S Pressman, “**Software Engineering**”, 2000 Edition, Dreamtech Publications
4. Karli Watson, Richard Anderson , “**Professional .NET 1.1**” , 2004 Edition, Wrox Publications
5. Pooja Bembey and Kuljit Kaur with NIIT,”**Microsoft Visual Studio.Net Professional Projects**”,2002 Edition, Prentice Hall of India Private Limited.

Websites:

1. www.msdn.microsoft.com
2. www.aspcity.com
3. www.aspdotnetheaven.com
4. www.codeguru.com

ANNEXURE

A. DFD

B. List of Tables

Table Name : BusDetails

Primary Key : BusNumber

Column Name	Datatype	Description
BusNumber	varchar(50)	Identifier for the bus
Starting_point	varchar(50)	The starting point of the bus route
ending_point	varchar(50)	The ending point of the bus route

Table Name : QueryTab

Primary Key : Query_no

Foreign Key : User_id

Column Name	Datatype	Description
Query_no	numeric(18, 0)	Unique identifier for each query
Query_heading	varchar(100)	Short title or heading for the query
Detailed_Query	varchar(MAX)	Detailed description of the query
Com_number	varchar(50)	Complaint number associated with the query
User_id	numeric(18, 0)	Identifier for the user who posted the query

Table Name : StopMaster

Primary Key : Stop_id

Foreign Key : BusNumber

Column Name	Datatype	Description
Stop_id	numeric(18, 0)	Unique identifier for each bus stop entry
BusNumber	varchar(50)	Identifier for the bus
Stop_no	numeric(18, 0)	Numeric order of the stop in the bus route
Stop_name	varchar(50)	Name of the bus stop
Arrival_time	varchar(50)	Scheduled arrival time at this stop
Charges	varchar(50)	Fare associated with reaching this stop

Table Name : userreg

Primary Key : Userid

Column Name	Datatype	Description
Userid	numeric(18, 0)	Unique identifier for each user
name_user	varchar(20)	Name of the user
gender	varchar(6)	Gender of the user
email_id	varchar(30)	Email ID of the user
mobile_number	numeric(10, 0)	Mobile number of the user
username	varchar(20)	Username for user login
password	varchar(15)	Password for user login

C. Sample Code

User Registration Coding

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

public partial class frmuserregistration : System.Web.UI.Page
{
    dbconnectDataContext db = new dbconnectDataContext();

    protected void Page_Load(object sender, EventArgs e)
    {
        if (IsPostBack == false)
        {
            var maxid = db.userregs.OrderByDescending(o => o.Userid).Select(s => new {
s.Userid }).Take(1);
            if (maxid.Count() <= 0)
            {
                txt_userid.Text = "1";
            }

            foreach (var item in maxid)
            {
                decimal str = item.Userid;
                txt_userid.Text = (str + 1).ToString();
            }
        }
    }

    protected void btnregister_Click(object sender, EventArgs e)
    {
        userreg ur = new userreg();
        ur.Userid = Convert.ToDecimal(txt_userid.Text);
        ur.name_user = txt_name.Text;
        ur.gender = drop_gender.Text;
        ur.email_id = txt_emailid.Text;
        ur.mobile_number = Convert.ToDecimal(txt_mobile.Text);
        ur.username = txt_username.Text;
        ur.password = txt_pwd.Text;
        db.userregs.InsertOnSubmit(ur);
        db.SubmitChanges();
        Response.Redirect("frmuserlogin.aspx");
    }
}
```

User Login Coding

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

public partial class frmuserlogin : System.Web.UI.Page
{
    dbconnectDataContext db = new dbconnectDataContext();

    protected void Page_Load(object sender, EventArgs e)
    {
        if (IsPostBack == false)
        {

        }
    }
    protected void btnlogin_Click(object sender, EventArgs e)
    {
        if (txtusername.Text != "")
        {
            var query = db.userregs.Where(x => x.username == txtusername.Text).Where(x =>
x.password == txtpassword.Text).FirstOrDefault();
            if (query == null)
            {
                Page.ClientScript.RegisterStartupScript(this.GetType(), "scriptkey",
"<script>alert('Invalid Login');</script>");
                return;
            }
            else
            {
                Session["userid"] = query.Userid;
                Session["username"] = query.username;
                Response.Redirect("frmusermenu.aspx");
            }
        }
    }
}
```

Admin Login Coding


```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

public partial class frmadminlogin : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {

    }

    protected void btnlogin_Click(object sender, EventArgs e)
    {
        if ((txtusername.Text == "admin") && (txtpassword.Text == "admin"))
        {
            Response.Redirect("frmadminmenu.aspx");
        }

        else
        {
            Page.ClientScript.RegisterStartupScript(this.GetType(), "scriptkey",
"<script>alert('Password Incorrect');</script>");
        }
    }
}

```

Finding Bus Route

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

public partial class frmBusRoute : System.Web.UI.Page
{

    dbconnectDataContext db = new dbconnectDataContext();

    protected void Page_Load(object sender, EventArgs e)
    {
        if (IsPostBack == false)
        {

```

```

    }
}
protected void btn_route_Click(object sender, EventArgs e)
{
    string busNumber = txt_busno.Text.Trim();

    // Query for GridView1 - Route Details
    var routeDetails = from stop in db.StopMasters
        where stop.BusNumber == busNumber
        orderby stop.Stop_no ascending
        select new
        {
            stop.Stop_no,
            stop.Stop_name,
            stop.Arrival_time,
            stop.Charges
        };

    GridView1.DataSource = routeDetails;
    GridView1.DataBind();

    // Query for GridView2 - Bus Details
    var busDetails = from bus in db.BusDetails
        where bus.BusNumber == busNumber
        select bus;

    GridView2.DataSource = busDetails;
    GridView2.DataBind();
}
}

```

Stop Track Details

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

public partial class Frm_Stoptrack : System.Web.UI.Page
{
    dbconnectDataContext db = new dbconnectDataContext();

    protected void Page_Load(object sender, EventArgs e)
    {

```

```

if (IsPostBack == false)
{
    var stopNames = db.StopMasters
        .Select(s => s.Stop_name)
        .Distinct()
        .OrderBy(name => name)
        .ToList();

    drop_stop.DataSource = stopNames;
    drop_stop.DataBind();
}
}
protected void btn_route_Click(object sender, EventArgs e)
{
    string selectedStopName = drop_stop.SelectedValue;

    // Query to find matching bus details where the stop name matches Starting_point or
    Ending_point
    var query = db.BusDetails
        .Where(bd => bd.Starting_point == selectedStopName || bd.ending_point ==
selectedStopName)
        .Select(bd => new
        {
            bd.BusNumber,
            bd.Starting_point,
            bd.ending_point
        });

    // Bind query result to GridView2
    GridView2.DataSource = query;
    GridView2.DataBind();

    // Query for GridView1 - Stop Match Details Sorted by Arrival Time
    var stopMatchDetails = from stop in db.StopMasters
        where stop.Stop_name == selectedStopName
        orderby Convert.ToDateTime(stop.Arrival_time) ascending
        select new
        {
            stop.BusNumber,
            stop.Stop_name,
            stop.Arrival_time,
            stop.Charges
        };

    GridView1.DataSource = stopMatchDetails;

```

```

        GridView1.DataBind();
    }
}

```

Stop and Fare Details

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

public partial class FrmStop : System.Web.UI.Page
{
    dbconnectDataContext db = new dbconnectDataContext();

    protected void Page_Load(object sender, EventArgs e)
    {
        if (IsPostBack == false)
        {
            var maxid = db.StopMasters.OrderByDescending(o => o.Stop_id).Select(s => new {
s.Stop_id }).Take(1);
            if (maxid.Count() <= 0)
            {
                txt_stopid.Text = "1";
            }

            foreach (var item in maxid)
            {
                decimal str = item.Stop_id;
                txt_stopid.Text = (str + 1).ToString();
            }
        }
    }
    protected void Button1_Click(object sender, EventArgs e)
    {
        StopMaster sm = new StopMaster();
        sm.Stop_id = Convert.ToDecimal(txt_stopid.Text);
        sm.BusNumber = txt_busno.Text;
        sm.Stop_no = Convert.ToDecimal(txt_Stopno.Text);
        sm.Stop_name = txt_stopname.Text;
        sm.Arrival_time = txt_arrivaltime.Text;
        sm.Charges = txt_charges.Text;
        db.StopMasters.InsertOnSubmit(sm);
        db.SubmitChanges();
    }
}

```

```

        Response.Redirect("FrmStop.aspx");
    }
    protected void Button2_Click(object sender, EventArgs e)
    {
        var bd = db.StopMasters.FirstOrDefault(record => record.Stop_id ==
Convert.ToDecimal(txt_stopid.Text));
        if (bd != null)
        {
            db.StopMasters.DeleteOnSubmit(bd);
            db.SubmitChanges();
        }
        Response.Redirect("FrmStop.aspx");
    }
    protected void GridView1_SelectedIndexChanged(object sender, EventArgs e)
    {

        txt_stopid.Text = GridView1.SelectedRow.Cells[0].Text;
        txt_busno.Text = GridView1.SelectedRow.Cells[1].Text;
        txt_Stopno.Text = GridView1.SelectedRow.Cells[2].Text;
        txt_stopname.Text = GridView1.SelectedRow.Cells[0].Text;
        txt_arrivaltime.Text = GridView1.SelectedRow.Cells[0].Text;
        txt_charges.Text = GridView1.SelectedRow.Cells[0].Text;
    }
}

```

Bus Details

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

public partial class frmBusDetails : System.Web.UI.Page
{
    dbconnectDataContext db = new dbconnectDataContext();

    protected void Page_Load(object sender, EventArgs e)
    {
        if (IsPostBack == false)
        {
            var maxid = db.BusDetails.OrderByDescending(o => o.auto_id).Select(s => new {
s.auto_id }).Take(1);
            if (maxid.Count() <= 0)

```

```

        {
            txt_autoid.Text = "1";
        }

        foreach (var item in maxid)
        {
            decimal str = item.auto_id;
            txt_autoid.Text = (str + 1).ToString();
        }
    }
}

protected void Button1_Click(object sender, EventArgs e)
{
    BusDetail bd = new BusDetail();
    bd.auto_id = Convert.ToDecimal(txt_autoid.Text);
    bd.BusNumber = txt_busno.Text;
    bd.Starting_point = txt_startingPoint.Text;
    bd.ending_point = txt_endingpoint.Text;
    db.BusDetails.InsertOnSubmit(bd);
    db.SubmitChanges();
    Response.Redirect("frmBusDetails.aspx");
}

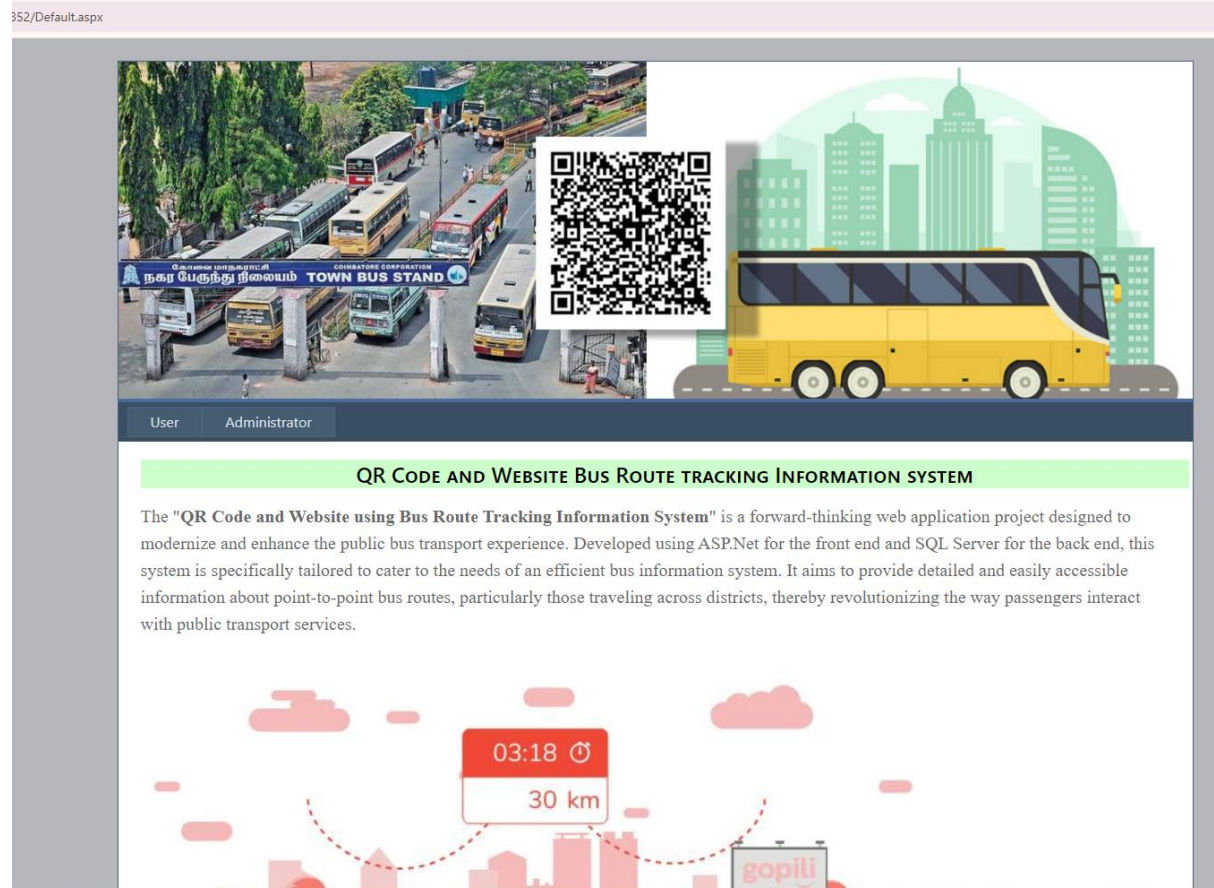
protected void Button2_Click(object sender, EventArgs e)
{
    var bd = db.BusDetails.FirstOrDefault(record => record.auto_id ==
Convert.ToDecimal(txt_autoid.Text));
    if (bd != null)
    {
        db.BusDetails.DeleteOnSubmit(bd);
        db.SubmitChanges();
    }
    Response.Redirect("frmBusDetails.aspx");
}

protected void GridView1_SelectedIndexChanged(object sender, EventArgs e)
{
    txt_autoid.Text = GridView1.SelectedRow.Cells[0].Text;
    txt_busno.Text = GridView1.SelectedRow.Cells[1].Text;
    txt_startingPoint.Text = GridView1.SelectedRow.Cells[2].Text;
    txt_endingpoint.Text = GridView1.SelectedRow.Cells[3].Text;
}
}

```


D. Screen Shots

Home Page



Admin Login

bst52352/frmadminlogin.aspx



The banner image features a photograph of a busy bus stand with several buses parked. Overlaid on the right side of the image is a large QR code and a stylized illustration of a yellow bus in front of a city skyline.

User Administrator

ADMIN LOGIN

Username

Password

Login




Admin Menu Page

2352/frmadminmenu.aspx



Bus Details

host:52352/frmBusDetails.aspx



[Home](#) [Bus Details](#) [Stop and Fare Details](#) [View Queries](#) [Registered Users](#) [Log out](#)

BUS DETAILS

Auto ID

274

Bus Number

Starting Point

Ending Point

Create

Delete

	Auto ID	Bus Number	Starting Point	Ending Point
Select	1	1	Maruthamalai	Avarampalayam
Select	2	1A	Ondipudur	Vadavalli
Select	3	1C	Ondipudur	Vadavalli
Select	4	1D	Ondipudur	Maruthamalai
Select	5	1E	Ondipudur	Maruthamalai
Select	6	2	Perur	Polytechnic
Select	7	2A	Perur	Polytechnic

Stop and Fare Details

localhost:52352/FrmStop.aspx



- Home
- Bus Details
- Stop and Fare Details
- View Queries
- Registered Users
- Log out

STOP AND FARE DETAILS

Stop Id

Bus Number

Stop Number

Stop Name

Arrival Time

Charges

CreateDelete

Stop_id	BusNumber	Stop_no	Stop_name	Arrival_time	Charges
1	62	1	Kovilpalayam Bus Stand	6:40 AM	20
2	62	2	S.S. Kulam	6:42 AM	20
3	62	3	V.J. Nagar	6:44 AM	20
4	62	4	Vaiyampalayam	6:51 AM	20
5	62	5	Ranganathar Polytechnic	7:00 AM	20
6	62	6	Athipalayam	7:01 AM	20
7	62	7	Idigarai	7:05 AM	20
8	62	8	Indian Overseas Bank	7:06 AM	20
9	62	9	Government High School Idigarai	7:08 AM	20
10	62	10	Maniakampalayam	7:12 AM	20
11	62	11	Cardwell Industry	7:13 AM	20
12	62	12	Sengalipalayam	7:16 AM	20
13	62	13	Vattamalaipalayam	7:18 AM	20
14	62	14	Ganapathy Colony	7:18 AM	20

View Queries

/frm_vwqueries.aspx






[Home](#) [Bus Details](#) [Stop and Fare Details](#) [View Queries](#) [Registered Users](#) [Log out](#)

VIEW QUERIES

Query_no	Query_heading	Detailed_Query	Com_number	User_id
1	Noise coming from the chair seats	So much noise coming from the chair seats. Maintenance is worst.	9842587963	1

Registered Users

2/frm_registeredusers.aspx



Home Bus Details Stop and Fare Details View Queries Registered Users Log out

REGISTERED USERS

User ID	Name	Gender	Email ID	Mobile Number	Username	Password	
1	Anand	Male	mailtoanandcbe@gmail.com	9843462104	12345	****	Edit Delete
5	Kishore	Male	kishore123@gmail.com	9843462104	kishore	****	Edit Delete

User Registration

t:52352/rfmuserregistration.aspx



User Administrator

USER REGISTRATION

User ID (Auto Generate)	<input type="text" value="5"/>
Name of the user	<input type="text" value="Kishore"/>
Gender	<input type="text" value="Male"/>
Email Id	<input type="text" value="kishore123@gmail.com"/>
Mobile Number	<input type="text" value="9843462104"/>
Username	<input type="text" value="kishore"/>
Password	<input type="password" value="*****"/>
<input type="button" value="Register"/>	

User Login

352/frmuserlogin.aspx



USER LOGIN


Username	<input type="text" value="kishore"/>
Password	<input type="password" value="*****"/>
	<input type="button" value="Login"/>

User Menu



Finding Bus Route

localhost:52352/frmBusRoute.aspx



[Home](#) [Bus Route](#) [Stop Details](#) [QR Scan](#) [Query](#) [Log out](#)

Bus ROUTE

Enter the Bus Number

Find Route and Charges

auto_id	BusNumber	Starting_point	ending_point
166	62	Railway Station	Vaiyampalayam

Stop_no	Stop_name	Arrival_time	Charges
1	Kovilpalayam Bus Stand	6:40 AM	20
2	S.S. Kulam	6:42 AM	20
3	V.J. Nagar	6:44 AM	20
4	Vaiyampalayam	6:51 AM	20
5	Ranganathar Polytechnic	7:00 AM	20
6	Athipalayam	7:01 AM	20
7	Idigarai	7:05 AM	20
8	Indian Overseas Bank	7:06 AM	20
9	Government High School Idigarai	7:08 AM	20
10	Maniakampalayam	7:12 AM	20
11	Cardwell Industry	7:13 AM	20
12	Sengalipalayam	7:16 AM	20
13	Vattamalaipalayam	7:18 AM	20

Finding Bus Arrival to Particular Stop

_Stoptrack.aspx



- Home
- Bus Route
- Stop Details
- QR Scan
- Query
- Log out

STOP DETAILS

Choose the Stop Name

▼

Find Bus and Timings Details

BusNumber	Starting_point	ending_point
12C	Eachanari	Polytechnic

BusNumber	Stop_name	Arrival_time	Charges
12D	Eachanari	7:39 AM	20

Generate QR Code – Bus information Details

mQRCode.aspx

The screenshot displays the 'mQRCode.aspx' web application. At the top, there is a header banner featuring a collage of images: a bus stand with several buses, a QR code, and a yellow bus against a city skyline background. Below the banner is a navigation menu with the following items: Home, Bus Route, Stop Details, QR Scan, Query, and Log out. The main content area is titled 'QR CODE' in a green bar. Below this, there is a form with the label 'Enter the Bus No' and a text input field containing the number '62'. A 'Get QR Code' button is positioned to the right of the input field. Below the button, a large QR code is generated and displayed.

User – Query form



Home Bus Route Stop Details QR Scan Query Log out

QUERY FORM

Query Number

1

Query Heading

Noise coming from the chair seats

Detailed Query

So much noise coming from the chair seats. Maintenance is worst.

Communication Number

9842587963

User ID

1

Post